

Graduado en Ingeniería Informática

Universidad Politécnica de Madrid

Facultad de Informática

TRABAJO FIN DE GRADO

Magnificador basado en teléfono móvil:
función de magnificación

Autor: Damian Jozef Trzpis

Director: José Luis Fuertes

MADRID, 13 ENERO DE 2013

ÍNDICE

1. INTRODUCCIÓN	1
2. ESTADO DE LA CUESTIÓN	2
2.1. MAGNIFICADORES.....	2
2.1.1. Magnificación del sistema operativo	2
2.1.1.1. Mobile Magnifier.....	2
2.1.1.2. Zoom.....	3
2.1.1.3. Zooms	4
2.1.2. Magnificación a través de la cámara.....	4
2.1.2.1. Magnify	5
2.1.2.2. ILoupe	5
2.1.2.3. Cozy Magnifier & Microscope +	6
2.1.3. Resumen de las características de los programas de magnificación	7
2.2. PROGRAMACIÓN ANDROID	7
2.2.1. ¿Qué es Android?	8
2.2.2. Arquitectura Android	8
2.2.2.1. Kernel de Linux	9
2.2.2.2. Bibliotecas	10
2.2.2.3. Entorno de ejecución.....	10
2.2.2.4. Marco de aplicación	11
2.2.2.5. Aplicaciones	12
2.3. OPENCV	12
2.3.1. Estructura modular de OpenCV	13
3. DESCRIPCIÓN DEL PROBLEMA	14
3.1. RECONOCIMIENTO DEL PROBLEMA.....	14
3.2. PLANTEAMIENTO DEL PROBLEMA	14
4. SOLUCIÓN.....	16
4.1. ANÁLISIS.....	16
4.1.1. Público al que va dirigido	16
4.1.2. Requisitos.....	16
4.1.2.1. Requisitos Generales:	16
4.1.2.2. Requisitos Específicos para la parte funcional:	17
4.1.3. Decisiones	17
4.2. DISEÑO.....	17
4.2.1. UML.....	17
4.2.2. Casos de uso y diagramas de secuencia.....	19
4.2.2.1. Modo de visualización RGB	19
4.2.2.2. Modo de visualización Gray	20
4.2.2.3. Modo de visualización BGR	20
4.2.2.4. Modo de visualización Blanco/Negro	21
4.2.2.5. Capturar la imagen	22
4.2.2.6. Dejar de capturar la imagen.....	23
4.2.2.7. Hacer zoom sobre una imagen capturada	24
4.2.2.8. Desplazarse sobre una imagen capturada	25
4.2.2.9. Activar el flash	26

4.2.2.10.	Desactivar el flash.....	27
4.2.2.11.	Activar auto enfoque.....	28
4.2.2.12.	Desactivar auto enfoque.....	29
4.2.2.13.	Activar macro enfoque.....	30
4.2.2.14.	Activar enfoque de área.....	31
4.2.2.15.	Activar enfoque de video.....	32
4.2.2.16.	Activar low fps.....	33
4.2.2.17.	Activar default fps.....	34
4.2.2.18.	Cambiar contraste.....	35
4.2.2.19.	Invertir colores.....	36
4.2.2.20.	Activar estabilización del video.....	37
4.2.2.21.	Desactivar estabilización del video.....	38
4.3.	IMPLEMENTACIÓN Y PRUEBAS.....	39
4.3.1.	<i>Pruebas de módulo cámara.....</i>	40
4.3.2.	<i>Pruebas de módulo librería.....</i>	42
4.3.3.	<i>Pruebas de sistema.....</i>	49
5.	RESULTADOS Y CONCLUSIONES.....	51
6.	FUTURAS LÍNEAS DE TRABAJO.....	53
7.	BIBLIOGRAFÍA.....	55
8.	ANEXO: MANUAL DE USUARIO DE LA LIBRERÍA.....	57
8.1.	PREPARACIÓN DEL ENTORNO DE TRABAJO.....	57
8.2.	CÓMO TRABAJAR.....	57

Resumen

En este Trabajo fin de Grado se ha creado la parte funcional de una aplicación móvil dejando la parte de la interfaz libre para su futura implementación. La aplicación es un magnificador y está destinada a aquellas personas con baja visión, permitiéndoles hacer aumento de imágenes y textos capturados por la cámara del dispositivo, con la posibilidad de que puedan cambiar algunos parámetros de la magnificación para poderlo ver mejor.

La parte funcional de la aplicación fue nombrada como librería magnificador, ya que se compone de métodos implementados en Java que permiten hacer diferentes modificaciones de la imagen o del video visualizado. Se cubrieron todas las fases de desarrollo más significativas de un sistema software: análisis, diseño, implementación y pruebas. La aplicación se desarrolló para Android. Se trabajó en Eclipse con Android SDK. Para el procesamiento de las imágenes se aprovechó una librería externa, OpenCV, para evitar “inventar la rueda”, es decir, no escribir algoritmos de transformación de las imágenes que seguramente no podrían ser tan eficientes como los implementados en dicha librería creada por Intel.

Summary

In this Final Project, the functional part of mobile phone application was created, leaving the interface part for future implementation. The application is a magnifier and is oriented to people with very low vision allowing them to enlarge images of printed documents captured by the camera of the mobile phone device, with the possibility that they can change some parameters of magnification so that it can see better.

The functional part of this application was named magnifier library because it composes of methods, implemented in Java, that allows changing between different modes of the preview of the image or video. All of the most significant phases of software development were satisfied: analysis, design, implementation and tests. The application was created for Android. The work was done in Eclipse with Android SDK plugin. For the image processing, an external library, OpenCv, was used, to avoid the unreachable intent of creation of effective algorithms that would never be so potent like the implemented in this library created by Intel.

1. Introducción

El propósito de este trabajo es crear un programa para un dispositivo móvil que ayude a las personas de baja visión, a leer un documento impreso. Hoy en día existen dispositivos específicos para tal fin, pero su precio es muy elevado, lo cual impide a la mayoría de los usuarios adquirirlos, o no son dispositivos portátiles. Gracias al desarrollo tecnológico del sector móvil, es posible crear un sistema móvil que sea igual de efectivo pero mucho más barato, aunque resultará necesario tener un dispositivo de gama media-alta que pueda soportar dicho sistema. Las ventajas de tener un programa magnificador en el móvil son numerosas, aparte del coste; por ejemplo, no es necesario llevar otro aparato que puede que no sea tan cómodo para transportar, o que requiera una mesa donde apoyarlo para que pueda estabilizar la imagen o incluso que necesite una pantalla adicional para transmitirla.

Para dar solución a esto, se ha pensado desarrollar un sistema completo que permita a un usuario magnificar las imágenes capturadas por una cámara de un dispositivo móvil para facilitar la lectura de documentos impresos por personas de baja visión. Dicho sistema se compondrá de una librería encargada de gestionar todas las operaciones de manejo de la cámara (enfoque, flash, estabilización, cambios de color, zoom, etc.) y de una interfaz de usuario que, haciendo uso de la librería, permitirá al usuario manejar el sistema de magnificación de manera cómoda e intuitiva. Este Trabajo de Fin de Grado se va a centrar en el desarrollo de la librería de magnificación.

Los objetivos del sistema son los siguientes:

1. Diseñar e implementar el sistema base de magnificación.
2. Diseñar e implementar los modos de alto contraste.
3. Diseñar e implementar el control del flash y la función macro.
4. Diseñar e implementar un sistema de estabilización de imagen.

2. Estado de la cuestión

2.1. Magnificadores

A continuación se describirán dispositivos o aplicaciones similares, que han servido de ejemplo de qué es lo que debe hacer el magnificador que se ha de desarrollar, qué errores no debe cometer o qué debe mejorar con respecto a los demás dispositivos o aplicaciones para que sea un sistema diferente, más efectivo, eficaz y accesible.

Las plataformas que se han tenido en cuenta eran: Android [Google, 2008], iOS [Apple, 2010] y Windows Phone [Microsoft, 2010].

Para hacer un estudio más específico se han dividido los magnificadores en dos grupos, el primero, magnificador del sistema operativo, lo que hacen es aumentar el tamaño del contenido de la pantalla y el segundo, magnificación a través de la cámara, como su nombre lo indica hace uso de la cámara del dispositivo para magnificar la imagen o video.

2.1.1. Magnificación del sistema operativo

El tipo de magnificadores que se nombrarán y explicarán a continuación, en general, lo que hacen es aumentar el tamaño del contenido de la pantalla, para que el usuario pueda tener una mejor visión. A diferencia del magnificador que se desarrollará, estos magnificadores no hacen uso de la cámara.

2.1.1.1. Mobile Magnifier

Es un magnificador a pantalla completa para teléfonos móviles que amplía y mejora la legibilidad del contenido de la pantalla, detecta las áreas de interés para hacerlas más visibles, y ofrece accesos directos para cambiar los ajustes sobre la marcha. En la figura 1, se puede observar el funcionamiento de la aplicación.



Figura 1: Ejemplo de pantalla magnificada con Mobile Magnifier.

Entre las principales características de Mobile Magnifier [Code Factory, 2013] están:

- Agranda contenido de la pantalla en 10 niveles de magnificación (de 1,25 x a 16x).
- Suaviza las curvas y los bordes de las fuentes y gráficos para facilitar la legibilidad.
- Dispone de 3 diseños diferentes: pantalla completa, *split*, y vista distribuida.
- Ofrece 7 esquemas de color diferentes para un mejor contraste visual incluyendo color normal, blanco y negro, escale de grises y los colores invertidos.
- Amplía la pantalla entera o sólo las áreas de interés.
- Cuenta con una función de desplazamiento automático, con cinco niveles de velocidad, que gira automáticamente el texto si no cabe en la pantalla cuando se magnifica.
- Proporciona varias teclas de acceso rápido para acceder a las funciones más comunes de la lupa.
- Incluye accesos directos especiales que le dan una vista ampliada de los iconos de estado, indicadores, así como la fecha y la hora.
- Puede ser configurado para iniciarse automáticamente cuando el teléfono móvil está encendido.
- Puede ser utilizado con o sin *Mobile Speak* [Code Factory, 2013].¹

Una de las desventajas que se pueden nombrar de este magnificador, es que aún no está disponible para algunos dispositivos móviles, como por ejemplo: Nokia C6-01, C7, E6, E7, N8 y X7.

2.1.1.2. Zoom

Zoom [Leibs, 2010] es una aplicación de magnificación de pantalla integrado en el sistema operativo de todos los Apple Mac OS X y productos iOS diseñados para hacer que los ordenadores sean más accesibles a las personas con discapacidad visual.

Zoom aumenta todo lo que aparece en pantalla - incluyendo texto, gráficos y vídeo - hasta 40 veces su tamaño original en máquinas Mac, y hasta 5 veces en los dispositivos iOS.

Los usuarios pueden activar el zoom a través de comandos de teclado, mover la rueda del ratón o usando gestos del *trackpad*, si se trata de un ordenador portátil, en el caso de los dispositivos móviles, tocando la pantalla con tres dedos.

¹ Mobile Speak es una línea de productos de lectores de pantalla, se compone de aplicaciones software instaladas en un teléfono móvil o PDA, que permiten a las personas ciegas utilizar el dispositivo, incluso si no pueden leer la pantalla visual [Wikipedia, 2012a].

Al utilizar Zoom en el iPad, iPhone e iPod Touch, aunque el rango de aumento (2x a 5x) es menor que en el caso que en un equipo Mac, se puede magnificar toda la pantalla y funciona a la perfección con cualquier aplicación.

Zoom ayuda a que sea más fácil leer el correo electrónico, escribir en un teclado pequeño, y acceder a aplicaciones de compra y gestión de la configuración.

Una vez utilizado Zoom, todos los gestos estándar iOS - tocar ligeramente y pellizcar, toque y rotor - aún funcionan cuando la pantalla se amplía.

Aunque parece una aplicación cuyo funcionamiento es eficaz y eficiente, Zoom no se puede utilizar en paralelo con el lector de pantalla *VoiceOver*, lo cual le resta accesibilidad. También cabe mencionar, que la mayoría de los productos Apple, no son los más baratos del mercado, por lo cual muchos de los usuarios no los pueden adquirir.

2.1.1.3. Zooms

Es un magnificador expresamente diseñado para pantallas pequeñas, como las de móviles. Zooms [Issue, 2007] permite:

- La pantalla se puede dividir, o utilizar completamente. En el primer caso, se mantienen las informaciones importantes en zonas fijas.
- Foco autoregurable. Se pueden mantener en la pantalla las líneas programadas: 1, 2, 3 etc. Zooms guarda siempre el mismo número de líneas en la pantalla.
- Los colores pueden configurarse: modo normal, blanco y negro o negativo, a gusto del usuario.
- Como todo no cabe en la pantalla cuando se amplía la imagen, la información puede moverse a velocidad definida por el usuario, con desplazamiento automático o mediante las flechas.
- Puede ser utilizado con Talks.²

2.1.2. Magnificación a través de la cámara

El tipo de magnificadores que se nombrarán y explicarán a continuación, utilizan la cámara del dispositivo para magnificar la imagen o video que se está observando, al igual que el magnificador que se desarrollará.

² Talks es un programa diseñado expresamente para ciegos, personas con problemas de visión, y para personas mayores. Desarrollado en Alemania, permite al móvil leer en voz alta la pantalla, ofreciendo al usuario con problemas visuales una autonomía total.

2.1.2.1. Magnify

Magnify [Appd Lab, 2011], es una aplicación de magnificación que realiza las siguientes funciones:

- Dos toques: Aumentar/disminuir
- Un toque: Enfoque
- Mantener presionado: Flash apagado/encendido
- Modo negativo

Aunque no presenta muchas opciones, este magnificador parece tener mucha fama en Google Play, ya que cuenta con más de cuatro mil usuarios. Algunos, incluso afirman, que gracias a su gran aumento e interacción simple, puede servir para las personas mayores o personas con baja visión.

Al probar la aplicación, funciona muy bien, pero con tan escasas funciones, cabe la duda de que si realmente puede resultar útil para las personas de baja visión, porque no sólo es cuestión de aumentar o disminuir una imagen tal como se muestra en la Figura 2, porque eso ya lo puede hacer el dispositivo móvil en sí, no haría falta tener una aplicación que haga eso, sino también de cambiar los modos de visualización porque a lo mejor es lo que necesita el usuario para poder ver correctamente como es el caso de las personas con daltonismo.



Figura 2: Un ejemplo de las capacidades de Magnify.

2.1.2.2. iLoupe

Es una aplicación de magnificación muy completa para iOS. iLoupe [insensiv, 2009] ofrece dos modos de ampliación: ampliación de video con calidad no muy buena y ampliación al tomar foto, hasta 22x con una calidad excelente. Sus características son:

- Magnificación video 8x/foto 22x.
- Estabilización video
- Guardar imagen

- Modo blanco y negro con nivel ajustable.

Desafortunadamente, al no disponer de un dispositivo iOS, no se ha podido probar la aplicación. Como ya se ha comentado antes, es una aplicación disponible para productos con sistema operativo iOS, cuyo precio elevado hace que no sean asequibles a todo usuario, lo que hace que esto sea una clara desventaja y si sumamos que tiene un coste de 0.99\$, muchos usuarios no la descargarían a pesar de no ser un precio muy alto, porque preferirían siempre una aplicación gratuita.

2.1.2.3. Cozy Magnifier & Microscope +

Cozy Magnifier & Microscope + [Google Play, 2012], es una aplicación para Android, orientada a personas con baja visión. En la Figura 3 se muestra un ejemplo de una imagen vista con la aplicación.

A continuación se muestra una lista de todas las opciones que posee, con sus características:

1. Magnificación
 - Modo lupa: Ampliación óptica (si el dispositivo lo ofrece) y digital.
 - Modo microscopio: la imagen es más clara y fina que usando el modo lupa.
 - Modo microscopio ofrece más zoom.
 - Enfoque automático continuo.
 - Barra de control para el zoom.
 - Tocar la pantalla para hacer zoom en todas partes.
2. Tomar foto/Capturar una imagen
 - Congelación de la pantalla para ver la imagen de forma estable.
 - Guardar una imagen, al tocar el icono de guardar.
 - Soporte WYSIWYG (*What You See Is What You Get* o lo que es lo mismo, lo que ves es lo que obtienes)
3. Efectos de imagen
 - Negativo.
 - Sepia.
5. Flash
 - Flash encendido/apagado
 - El botón designado para bajar el volumen, es el utilizado para encender-apagar el flash.
6. Visor de imágenes
 - Visualización rápida de imágenes para guardar la imagen.
 - Imagen de uso compartido.
 - Rápida visualización de imagen para las fotos.
7. Pantalla completa

- El botón de menú, para colocar la pantalla completa.



Figura 3: Ejemplo de imagen antes y después de utilizar la aplicación.

Si habría que encontrar alguna desventaja a esta aplicación bastante completa, sería que es una aplicación de pago y al tener muchas funciones y utilizar botones que ya tienen otras funciones predeterminadas, puede ser bastante confuso para el usuario.

2.1.3. Resumen de las características de los programas de magnificación

Una vez conocido algunos de los programas magnificadores que conforman el mercado actual, está bastante claro que éstos aunque cumplen con su objetivo, no han sido diseñados principalmente para las personas con baja visión, bien sea por sus insuficientes funciones o por la compleja interacción de la aplicación. Aunque algunas, incluso, podrían servir para éste grupo de personas, resultan que son aplicaciones de pago, que aunque no sea muy excesivo, siempre limita su instalación.

De las aplicaciones magnificadoras, la que más se asemeja al magnificador que se desarrollará son: Magnify y Cozy Magnifier & Microscope +.

Magnify es una aplicación cuya interacción es muy simple, pero no ofrece opciones de modos de visualización, enfoque, captura de imagen, etc.

Por otro lado, Cozy Magnifier & Microscope +, ofrece más opciones y modos de aumento y visualización, es un programa muy completo pero realmente no está destinado principalmente a personas con baja visión, además de no ser una aplicación gratuita.

2.2. Programación Android

Una limitación en la construcción de aplicaciones para móviles, es la arquitectura del móvil; Android, IOS o Windows Phone. El incuestionable ganador en la conquista del mercado es Android [Wikipedia, 2013]. Su código abierto ha permitido su gran

desarrollo y dominación en la mayoría de los móviles. La única pega es la calidad. Programar en Android puede hacerlo cualquiera, gratuitamente. Por lo tanto se ha creado un numero de aplicaciones mucho mayor que en el caso de IOS, donde hay que pagar para subir la aplicación.

No obstante, el objetivo de este proyecto es llegar al mayor número de usuarios y evitar costes, por lo que se ha seleccionado Android como la plataforma en la cual se va a desarrollar la aplicación.

2.2.1. ¿Qué es Android?

Android es un sistema operativo inicialmente diseñado para teléfonos móviles, aunque *tablets*, *netbooks*, ordenadores, portátiles, auriculares, Google TV, y demás dispositivos, ya utilizan dicho sistema, el cual está basado en un sistema operativo libre, gratuito y multiplataforma como lo es Linux 2.6 [Cruz., 2008].

Se podría decir que por excelencia, Android es uno de los sistemas operativos más demandados en la actualidad, debido a las facilidades de uso, no sólo para el usuario del dispositivo sino también para el desarrollador de aplicaciones. Al ser un sistema operativo completamente libre, fomenta tanto a los fabricantes para lanzar un teléfono cuyo coste sea más bajo que los demás dispositivos de la competencia, como a los programadores a desarrollar innumerables aplicaciones sin coste alguno, ya que para incluir sus aplicaciones no tienen que pagar absolutamente nada.

En cuanto a la experiencia del usuario, éste siente que puede aprovechar al máximo su dispositivo móvil, gracias a la amplia gama de aplicaciones que le ofrece Android, por lo anteriormente comentado. Y no sólo en este caso es importante la implicación del usuario con su dispositivo, sino que también si éste es lo suficientemente inquieto y le gustaría indagar aún más, el usuario puede acceder al código fuente y a la listas de incidencias para poder conocer cuáles son los problemas que no han sido resueltos y también cómo podría contribuir a reportar problemas nuevos, para que en versiones posteriores éstos puedan ser reparados.

2.2.2. Arquitectura Android

Una vez conocido qué es Android y los motivos por lo que gusta tanto a fabricantes, desarrolladores y usuarios, es momento de profundizar un poco más en el tema, es decir, saber cuáles son los 5 principales componentes del sistema operativo: *kernel* de Linux, l bibliotecas, entorno de ejecución, marco de aplicación y aplicaciones.

En la Figura 4, se puede observar cómo la arquitectura de Android está agrupada por capas [Vico, 2012], en la cual cada capa superior depende de la inmediatamente inferior para poder realizar sus funciones, motivo por el cual se denomina pila. Este modo de

estructuración del sistema operativo Android fue pensado así, para que el desarrollador de las aplicaciones no se vea en la necesidad de programar a bajo nivel las funcionalidades necesarias para que la aplicación pueda acceder o hacer uso de los componentes hardware del dispositivo móvil.

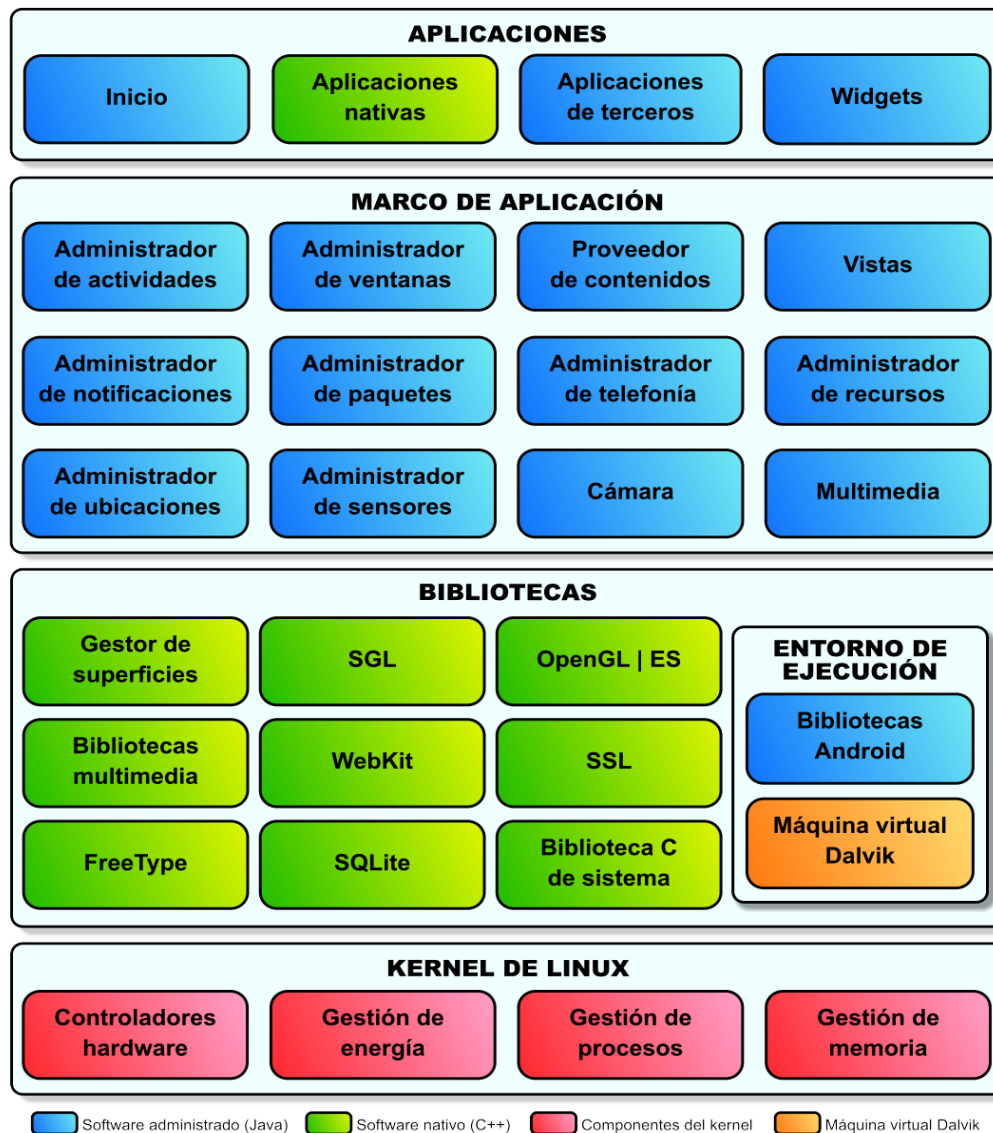


Figura 4: Arquitectura Android.

2.2.2.1. Kernel de Linux

El *Kernel* de Linux, es lo que está considerado como el núcleo o corazón del sistema operativo; permite que el software y hardware trabajen juntos.

Entre las principales funciones por las cuales Android depende de Linux 2.6, se pueden destacar las siguientes: administrar la energía del dispositivo, gestionar los procesos, gestionar la memoria para todos los programas y procesos en ejecución, y controlar los componentes hardware que permiten que se pueda acceder a los mismos sin necesidad de conocer el modelo o características precisas de los que están instalados en cada teléfono.

2.2.2.2. Bibliotecas

Las bibliotecas suelen estar escritas en los lenguajes C o C++ y compiladas e instaladas en cada dispositivo móvil antes de ponerlo a la venta, de ésta tarea se encarga el fabricante. Entre las bibliotecas que se incluyen normalmente se pueden destacar las siguientes:

- **Gestor de superficies (*Surface Manager*):** compone las imágenes que se muestran en la pantalla a partir de capas gráficas de dos y tres dimensiones.
- **SGL (*Scalable Graphics Library*):** es el motor gráfico de dos dimensiones de Android.
- **Bibliotecas multimedia:** contiene todos los códec necesarios visualizar, reproducir e incluso grabar numerosos formatos de imagen, vídeo y audio.
- **WebKit:** motor web utilizado por el navegador.
- **SSL (*Secure Sockets Layer*):** proporciona seguridad al acceder a Internet por medio de criptografía haciendo uso del protocolo SSL.
- **FreeType:** permite trabajar de manera eficaz con distintas tipografías.
- **SQLite:** motor de bases de datos relacionales.
- **Biblioteca C de sistema (libc):** proporciona funcionalidad básica para la ejecución de las aplicaciones.

2.2.2.3. Entorno de ejecución

El entorno de ejecución de Android no puede ser considerado como una capa como tal, ya que necesita de las bibliotecas que han sido nombradas anteriormente y a su vez también está formado por otras bibliotecas Android, por lo que vendría siendo una subcapa de las bibliotecas, si se pudiese llamar así.

La máquina virtual Dalvik, es el componente más importante del entorno de ejecución ya que es el encargado de ejecutar todas las aplicaciones que no están por defecto en Android. A pesar de que las aplicaciones en su mayoría están realizadas con Java, para compilarlas y ejecutarlas se utiliza la máquina virtual Dalvik, porque es más eficiente, es decir, sólo basta con ejecutarlo una única vez y se tiene la certeza de que funciona en cualquier dispositivo con sistema operativo Android.

2.2.2.4. Marco de aplicación

El marco de aplicación está conformado por todas las clases y servicios que utilizaran todas las aplicaciones que se encuentran en la capa que está inmediatamente encima, apoyándose en las bibliotecas nativas o por defecto de Android así como las que están en el entorno de ejecución.

Muchos de los componentes del marco de aplicación son más bibliotecas que acceden a los recursos a través de la máquina virtual Dalvik. Entre las más importantes se encuentran las siguientes:

- **Administrador de actividades (*Activity Manager*):** controla el ciclo de vida de las actividades.
- **Administrador de ventanas (*Windows Manager*):** organiza lo que se muestra en la pantalla.
- **Proveedor de contenidos (*Content Provider*):** encapsula los datos que serán compartidos por las aplicaciones, sin perder el control sobre cómo acceder a la información.
- **Vistas (*Views*):** contiene una amplia gama de vistas para poder construir la interfaz del usuario.
- **Administrador de notificaciones (*Notification Manager*):** notifica al usuario si alguna aplicación requiere su atención.
- **Administrador de paquetes (*Package Manager*):** gestiona y da información de los paquetes instalados en el dispositivo o los nuevos que se instalarán.
- **Administrador de telefonía (*Telephony Manager*):** permite realizar llamadas o enviar y recibir SMS/MMS, aunque no permite reemplazar o eliminar la actividad que se muestra cuando una llamada está en curso.
- **Administrador de recursos (*Resource Manager*):** proporciona acceso a todos los elementos propios de una aplicación que se incluyen directamente en el código. Permitiendo gestionar las aplicaciones, fuera del código, de modo que el usuario puede modificarlo según sus preferencias
- **Administrador de ubicaciones (*Location Manager*):** determina la posición geográfica del dispositivo Android y trabaja con mapas.
- **Administrador de sensores (*Sensor Manager*):** gestiona todos los sensores hardware disponibles en el dispositivo Android.
- **Cámara:** proporciona acceso a las cámaras del dispositivo Android, tanto para tomar fotografías como para grabar vídeo.
- **Multimedia:** conjunto de bibliotecas que permiten reproducir y visualizar audio, vídeo e imágenes en el dispositivo.

El componente que más interesa en la realización de este proyecto es la cámara ya que es la base de la aplicación.

2.2.2.5. Aplicaciones

La capa superior de esta pila software la forman, las aplicaciones, en la cual se incluyen todas las aplicaciones del dispositivo, valga la redundancia, tanto las que tienen interfaz de usuario como las que no, bien sea las que vienen de serie con el dispositivo como las instaladas por el usuario.

Lo principal a tener en cuenta de esta arquitectura es que todas las aplicaciones, utilizan el mismo marco de aplicación para acceder a los servicios que proporciona el sistema operativo. Esto implica dos cosas: que pueden crearse aplicaciones que usen los mismos recursos que usan las aplicaciones nativas (nada está reservado o inaccesible) y que pueden remplazarse cualquiera de las aplicaciones del teléfono por otra a elección del usuario. Este es el verdadero potencial de Android y lo que lo diferencia de su competencia: control total por parte del usuario del software que se ejecuta en su teléfono.

2.3.OpenCV

OpenCV es una librería específica para visión artificial, desarrollada por Intel [Wikipedia, 2012b]. Desde que apareció su primera versión alfa, en el mes de enero de 1999, se ha utilizado en infinidad de aplicaciones, desde sistemas de seguridad con detección de movimiento, hasta aplicaciones de control de procesos donde se requiere reconocimiento de objetos. Esto se debe a que su publicación se da bajo licencia BSD, que permite que sea usada libremente para propósitos comerciales y de investigación con las condiciones en ella expresadas

OpenCV es multiplataforma, existiendo versiones para GNU/Linux, Mac OS X y Windows. Contiene más de 500 funciones que abarcan una gran gama de áreas en el proceso de visión, como reconocimiento de objetos (reconocimiento facial), calibración de cámaras, visión estéreo y visión robótica.

La programación se realizó en código C y C++ optimizados, aprovechando además las capacidades que proveen los procesadores multinúcleo. OpenCV puede además utilizar el sistema de primitivas de rendimiento integradas de Intel, un conjunto de rutinas de bajo nivel específicas para procesadores Intel. La versión de OpenCV para Android tiene una interfaz en Java [Itseez, 2012].

Por ser una librería que permite tanta versatilidad, por tener una amplia gama de funciones sobre todo en temas de visión, será la que se utilizará en conjunto con Android para el desarrollo de la aplicación magnificador.

2.3.1. Estructura modular de OpenCV

OpenCV tiene una estructura modular, lo que significa que el paquete incluye varias bibliotecas compartidas o estática [Unknown, 2012a]. Los módulos disponibles son los siguientes:

- **core:** un módulo compacto que define las estructuras de datos básicas, incluyendo la matriz densa multidimensional Mat y las funciones básicas utilizadas por todos los otros módulos.
- **imgproc:** un módulo de procesamiento de imagen que incluye la imagen lineal y no lineal de filtrado, transformaciones geométricas imagen, conversión de espacio de color, histogramas, y así sucesivamente.
- **video:** un módulo de análisis de vídeo que incluye estimación del movimiento, la sustracción de fondo, y los algoritmos de seguimiento de objetos.
- **calib3d:** básicas de múltiples algoritmos de visión geometría, la calibración de una cámara y de música, objeto plantear la estimación, los algoritmos de correspondencia estéreo, y elementos de reconstrucción 3D.
- **features2d:** detector de características sobresalientes, descriptores y comparadores descriptores.
- **objdetect:** detección de objetos e instancias de las clases predefinidas (por ejemplo, caras, ojos, tazas, gente, coches, etc.).
- **highgui:** un fácil de usar interfaz de video *codecs* captura, imagen y vídeo, así como las capacidades de interfaz de usuario sencilla.
- **gpu:** acelerados por GPU algoritmos de diferentes módulos de OpenCV.

3. Descripción del problema

3.1.Reconocimiento del problema

Como se ha comentado anteriormente, existe la necesidad de tener un sistema de visualización, a través de la cámara del teléfono móvil, que permita aumentar una imagen, y visualizarla de diferentes modos, para que a las personas con baja visión, les sea más fácil ver las imágenes.

Este Trabajo de Fin de Grado, estará dedicado a la creación del núcleo de la aplicación, es decir, la parte funcional del sistema. Su objetivo es crear servicios, que permitan el manejo eficaz de videos e imágenes, que puedan ser fácilmente utilizados a la hora de desarrollar la interfaz de la aplicación.

3.2. Planteamiento del problema

El objetivo del presente proyecto es realizar un magnificador para dispositivos móviles, orientado principalmente para las personas que tienen baja visión, aunque puede ser utilizado por cualquier persona que no tenga ningún tipo de discapacidad.

El programa magnificador hará lo siguiente:

- **Función de magnificación digital:** es la función base obligatoria, es decir, es la que se encarga del aumento o disminución de las imágenes o del video.
- **Capturar una imagen:** es una función que es necesaria implementar para que el usuario pueda hacer uso del desplazamiento sobre una imagen fija, así como también el zoom sobre dicha imagen. También es muy útil porque no todas las personas tienen un buen pulso, y para aquéllos que les tiembla mucho la mano y desean tener una imagen fija de lo que están observando es una buena opción. Es importante destacar que la captura de la imagen no debe confundirse con tomar una foto, ya que la captura no se almacena en el álbum del dispositivo.
- **Desplazarse sobre una imagen fija:** es la típica función que se utiliza tras capturar una imagen y hacer zoom sobre ella, para poder observar toda la imagen en su totalidad.
- **Mostrar distintos modos de visualización:** entre los modos de visualización que se quieren implementar destacan: RGB, BGR, escala de grises, blanco y negro, contraste ajustable, imagen binaria con colores seleccionables e invertir colores. La razón por la cual se consideró cambiar los modos de visualización, es que muchas personas con baja visión necesitan cambiar los colores, los tonos o el contraste para poder visualizar mejor la imagen o video.
- **Activar o desactivar el flash:** básicamente servirá para dar más luminosidad a lo que se está observando, sobre todo para los lugares oscuros; podría ser una función muy útil para el usuario.

- **Activar distintos tipos de enfoque:** entre los distintos tipos de enfoques que se implementaran están: auto, macro, video, área y sin enfocar. Permitirá al usuario tener distintos tipos de enfoque dependiendo de la situación o condición en la que se encuentren.
- **Reducir la cantidad de imágenes por segundo (fps) mostradas:** disminuye el “temblor” de la imagen cuando se usa un nivel de magnificación grande ya que no se muestran tantas imágenes por lo tanto el documento observado parece más estable.
- **Estabilizar el video:** servirá para eliminar “temblor” de la imagen cuando el móvil se mantiene en las manos y no está apoyado sobre algo fijo.

Este proyecto se desarrollará en dos partes: la primera consiste en la creación de una librería que implementa funciones de magnificación y la segunda consiste en el diseño de la interfaz de usuario adecuada que haga uso de dicha librería. Este Trabajo de Fin de Grado se centrará en la creación de una librería que implemente las funciones de magnificación que han sido explicadas previamente.

4. Solución

Para llevar a cabo la realización del magnificador, de una manera más eficaz y eficiente y cuyo coste sea menor al de los demás dispositivos o aplicaciones que hay en el mercado, que no son asequibles a todos los usuarios, se ha decidido programar en Android con ayuda de las librerías de OpenCV, ya que es código abierto, ofrece gran cantidad de funciones de procesamiento de imágenes que pueden ser aprovechadas si se quisiera ampliar la aplicación con otros servicios.

4.1. Análisis

4.1.1. Público al que va dirigido

Una vez identificado el problema, es importante dejar claro, el público al que va dirigida dicha aplicación para el teléfono móvil. Como ya se ha mencionado, principalmente los usuarios del sistema final serán personas con baja visión; no se estima el rango de edades de los usuarios, porque la falta de visión afecta personas de cualquier edad. Por lo tanto, la implementación de la aplicación debe ser lo más intuitiva posible, porque no todas las personas y mucho menos con distintas edades tienen las mismas capacidades de aprendizaje.

También es importante destacar el destinatario de este Trabajo, que es la persona que será responsable de desarrollar la parte interactiva de la aplicación, con lo cual es necesario garantizar la fácil asimilación de este trabajo, sobre todo del código fuente a entregar.

4.1.2. Requisitos

Los requisitos que se presentan en este apartado se han elaborado a partir de los objetivos anteriormente presentados y las reuniones con los profesores. Se intentó cubrir la mayoría de las necesidades de los posibles usuarios.

4.1.2.1. Requisitos Generales:

- R1. Aplicación móvil Android.
- R2. La aplicación tendrá opción de “capturar la imagen”.
- R3. La aplicación tendrá opción de activar/desactivar el autofocus.
- R4. La aplicación tendrá opción de aumentar el video/imagen.
- R5. La aplicación tendrá opción de activar/desactivar el flash.
- R6. La aplicación tendrá opción de activar/desactivar el enfoque macro.
- R7. La aplicación tendrá opción de activar el modo de visualización “blanco y negro”.
- R8. La aplicación tendrá opción de activar el modo de visualización “escala de grises”.

- R9. La aplicación tendrá opción de activar el modo de visualización “RGB”.
- R10. La aplicación tendrá opción de activar el modo de visualización “BGR”.
- R11. La aplicación tendrá opción de activar el modo de visualización colores invertidos (modo alto contraste).
- R12. La aplicación tendrá opción de incrementar el contraste de video/imagen.
- R13. La aplicación tendrá opción de activar/desactivar la estabilización de la imagen según disponibilidad.

4.1.2.2. Requisitos Específicos para la parte funcional:

- E1. Dejar el proyecto abierto para futura fusión con la parte de la interfaz. El proyecto y el código tienen que estar bien organizados y estructurados para poder ser utilizados posteriormente con facilidad de manera lógica.
- E2. Ofrecer una demostración de la aplicación, donde se puedan probar todas las opciones aunque no sea mediante una interfaz “cómoda” o intuitiva.
- E3. Ofrecer un manual de usuario de la librería para el programador de la interfaz.

4.1.3. Decisiones

Se ha tomado la decisión que el proyecto se hará en Android ya que es software libre y tanto el desarrollo como la publicación de las aplicaciones no conlleva ningún costo. El código fuente del proyecto será en Java. Se va a trabajar en Eclipse utilizando librerías para programación Android. Para las transformaciones de las imágenes se utilizarán librerías de OpenCV4Android.

4.2.Diseño

4.2.1. UML

La Figura 5 el diagrama UML en el cual se va a basar la aplicación del Magnificador. Como se puede observar en el mismo, está conformado principalmente por tres grandes clases: MagnificadorActivity, MagnificadorProcess y MagnificadorBase.

La clase MagnificadorActivity, que es la clase que será la primera en ejecutarse, cuando se comienza a utilizar la aplicación, creará una instancia de la clase MagnificadorProcess que extiende de la clase MagnificadorBase.

La clase MagnificadorBase, es donde se crearán las funciones propias de la librería de la aplicación. Esta clase tendrá dos módulos: módulo cámara, en el cual se hará la inicialización de la librería OpenCV y de la cámara y modulo librería donde se encontrarán los métodos que permitirán modificar la visualización. Como se puede

observar dicha clase extenderá de la clase SurfaceView, lo cual será necesario cuando entre en funcionamiento el uso de la cámara en la aplicación.

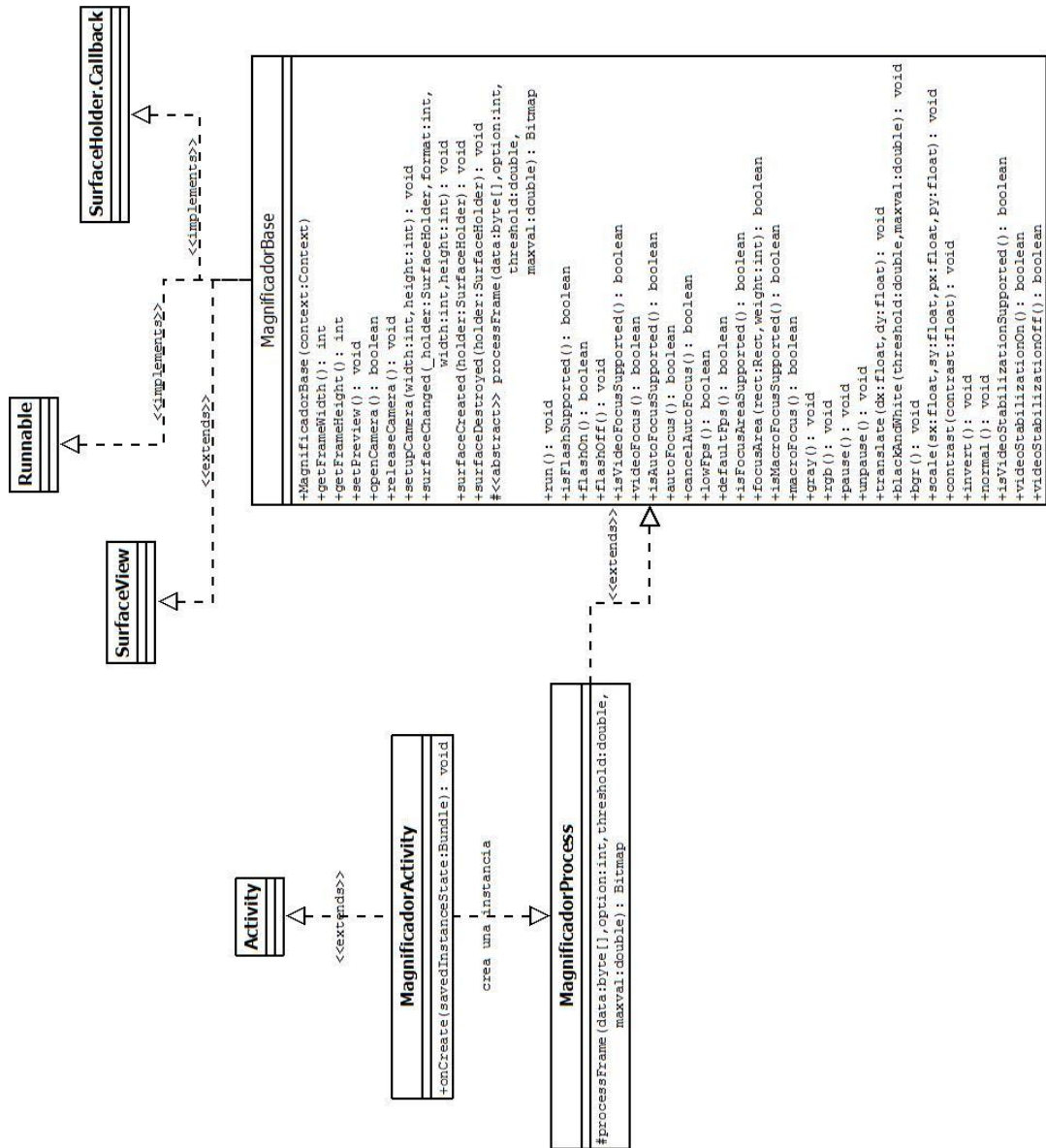


Figura 5: UML del Magnificador.

MagnificadorBase implementará a su vez dos clases más:

- SurfaceHolder.Callback será necesario para poder controlar el SurfaceHolder que contendrá la vista actual³

³ La vista actual representa el bloque de construcción básico para los componentes de la interfaz de usuario. Una vista ocupa un área rectangular en la pantalla y es responsable del dibujo y manejo de eventos. La vista es la clase base para los *widgets*, que se utilizan para crear componentes interactivos de la interfaz de usuario (botones, campos de texto, etc.)

- Runnable será necesario para crear el *thread* o hilo que va a ocuparse del procesamiento en tiempo real de la imagen

La clase `MagnificadorProcess` es en la que se implementará la función `processFrame` que será el método responsable del procesamiento del *frame* actual.

4.2.2. Casos de uso y diagramas de secuencia

4.2.2.1. Modo de visualización RGB

Descripción en formato breve

El usuario desea cambiar el modo de visualización actual a modo de visualización RGB.

Descripción en formato completo esencial

- Actor Principal
Usuario
- Actores Involucrados e Intereses
Usuario: Es la persona que tiene baja visión.
- Precondiciones
 1. La aplicación ha sido ejecutada.
 2. El dispositivo móvil posee una cámara trasera.
 3. El dispositivo móvil tiene sistema operativo Android 4.0.
 4. El dispositivo tiene instalado OpenCV Manager.
- Garantías de Éxito (Post condiciones)
 1. La aplicación cambia el modo de visualización.
- Escenario Principal de Éxito
 1. El usuario elige la opción “RGB”.
- Extensiones o Flujos Alternativos
No se consideran.
- Temas Abiertos
No se consideran.
- Diagrama de Secuencia: figura 6

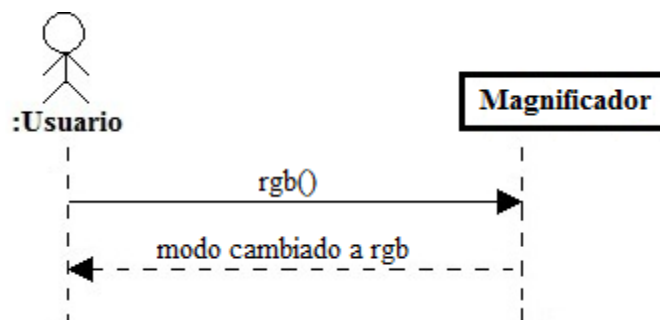


Figura 6: Diagrama de secuencia: Modo de visualización RGB.

4.2.2.2. Modo de visualización Gray

Descripción en formato breve

El usuario desea cambiar el modo de visualización actual a modo de visualización Gray.

Descripción en formato completo esencial

- Actor Principal
Usuario
- Actores Involucrados e Intereses
Usuario: Es la persona que tiene baja visión.
- Precondiciones
 1. La aplicación ha sido ejecutada.
 2. El dispositivo móvil posee una cámara trasera.
 3. El dispositivo móvil tiene sistema operativo Android 4.0.
 4. El dispositivo tiene instalado OpenCV Manager.
- Garantías de Éxito (Post condiciones)
 1. La aplicación cambia el modo de visualización.
- Escenario Principal de Éxito
 1. El usuario elige la opción “Gray”.
- Extensiones o Flujos Alternativos
No se consideran.
- Temas Abiertos
No se consideran.
- Diagrama de Secuencia: figura 7

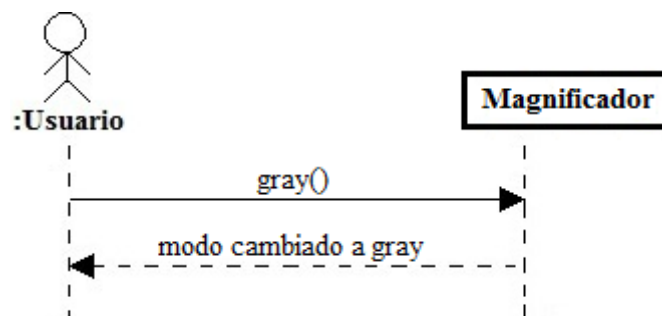


Figura 7: Diagrama de secuencia: Modo de visualización Gray.

4.2.2.3. Modo de visualización BGR

Descripción en formato breve

El usuario desea cambiar el modo de visualización actual a modo de visualización BGR.

Descripción en formato completo esencial

- Actor Principal
Usuario
- Actores Involucrados e Intereses
Usuario: Es la persona que tiene baja visión.
- Precondiciones
 1. La aplicación ha sido ejecutada.
 2. El dispositivo móvil posee una cámara trasera.
 3. El dispositivo móvil tiene sistema operativo Android 4.0.
 4. El dispositivo tiene instalado OpenCV Manager.
- Garantías de Éxito (Post condiciones)
 1. La aplicación cambia el modo de visualización.
- Escenario Principal de Éxito
 1. El usuario elige la opción “BGR”.
- Extensiones o Flujos Alternativos
No se consideran.
- Temas Abiertos
No se consideran.
- Diagrama de Secuencia: figura 8

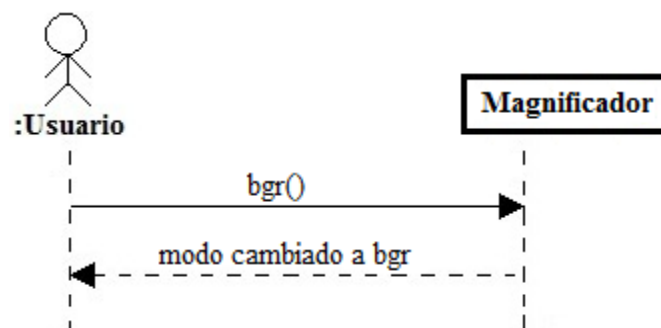


Figura 8: Diagrama de secuencia: Modo de visualización BGR.

4.2.2.4. Modo de visualización Blanco/Negro

Descripción en formato breve

El usuario desea cambiar el modo de visualización actual a modo de visualización Blanco/Negro.

Descripción en formato completo esencial

- Actor Principal
Usuario

- Actores Involucrados e Intereses
Usuario: Es la persona que tiene baja visión.
- Precondiciones
 1. La aplicación ha sido ejecutada.
 2. El dispositivo móvil posee una cámara trasera.
 3. El dispositivo móvil tiene sistema operativo Android 4.0.
 4. El dispositivo tiene instalado OpenCV Manager.
- Garantías de Éxito (Post condiciones)
 1. La aplicación cambia el modo de visualización.
- Escenario Principal de Éxito
 1. El usuario elige la opción “BW”.
- Extensiones o Flujos Alternativos
No se consideran.
- Temas Abiertos
No se consideran.
- Diagrama de Secuencia: figura 9

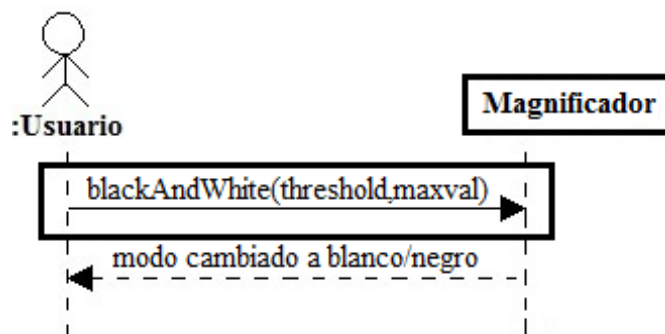


Figura 9: Diagrama de secuencia: Modo de visualización Blanco/Negro.

4.2.2.5. Capturar la imagen

Descripción en formato breve

El usuario desea capturar la imagen para poder tener una imagen fija de lo que está observando.

Descripción en formato completo esencial

- Actor Principal
Usuario
- Actores Involucrados e Intereses
Usuario: Es la persona que tiene baja visión.
- Precondiciones
 1. La aplicación ha sido ejecutada.

- 2. El dispositivo móvil posee una cámara trasera.
- 3. El dispositivo móvil tiene sistema operativo Android 4.0.
- 4. El dispositivo tiene instalado OpenCV Manager.
- Garantías de Éxito (Post condiciones)
 - 1. La aplicación captura la imagen.
- Escenario Principal de Éxito
 - 1. El usuario elige la opción “pause”.
- Extensiones o Flujos Alternativos

No se consideran.
- Temas Abiertos

No se consideran.
- Diagrama de Secuencia: figura 10

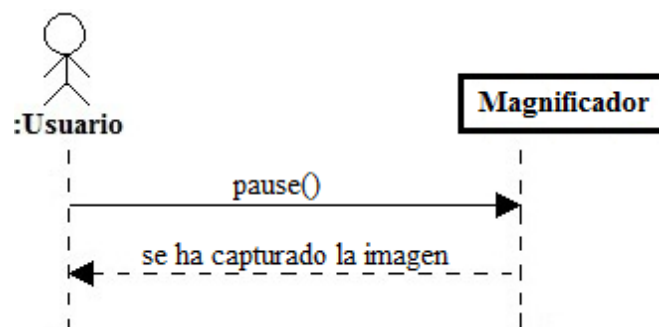


Figura 10: Diagrama de secuencia: Capturar la imagen.

4.2.2.6. Dejar de capturar la imagen

Descripción en formato breve

El usuario desea dejar de capturar la imagen para poder visualizar todo lo que está observando.

Descripción en formato completo esencial

- Actor Principal

Usuario
- Actores Involucrados e Intereses

Usuario: Es la persona que tiene baja visión.
- Precondiciones
 - 1. La aplicación ha sido ejecutada.
 - 2. El dispositivo móvil posee una cámara trasera.
 - 3. El dispositivo móvil tiene sistema operativo Android 4.0.
 - 4. El dispositivo tiene instalado OpenCV Manager.
- Garantías de Éxito (Post condiciones)

1. La aplicación deja de capturar la imagen.
- Escenario Principal de Éxito
 1. El usuario elige la opción “unpause”.
- Extensiones o Flujos Alternativos

No se consideran.
- Temas Abiertos

No se consideran.
- Diagrama de Secuencia: figura 11

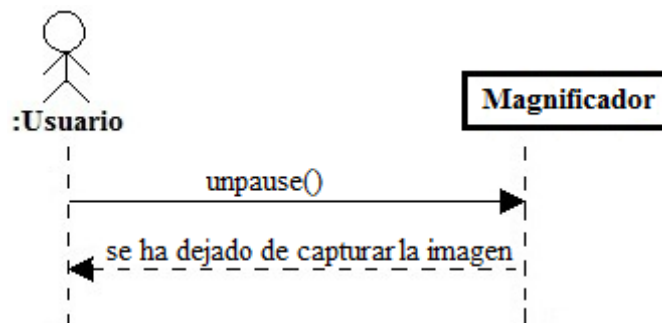


Figura 11: Diagrama de secuencia: Dejar de congelar la imagen.

4.2.2.7. Hacer zoom sobre una imagen capturada

Descripción en formato breve

El usuario desea hacer zoom de una imagen fija.

Descripción en formato completo esencial

- Actor Principal

Usuario
- Actores Involucrados e Intereses

Usuario: Es la persona que tiene baja visión.
- Precondiciones
 1. La aplicación ha sido ejecutada.
 2. El dispositivo móvil posee una cámara trasera.
 3. El dispositivo móvil tiene sistema operativo Android 4.0.
 4. El dispositivo tiene instalado OpenCV Manager.
- Garantías de Éxito (Post condiciones)
 1. El usuario hace zoom de una imagen fija.
 2. La aplicación aumenta o disminuye, según desee el usuario, la imagen fija.
- Escenario Principal de Éxito
 1. El usuario elige la opción “zoom”.
- Extensiones o Flujos Alternativos

No se consideran.

- Temas Abiertos

No se consideran.

- Diagrama de Secuencia: figura 12

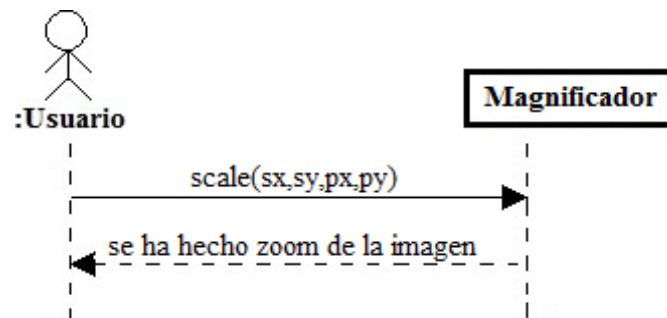


Figura 12: Diagrama de secuencia: Hacer zoom sobre una imagen capturada.

4.2.2.8. Desplazarse sobre una imagen capturada

Descripción en formato breve

El usuario desea desplazarse sobre una imagen fija.

Descripción en formato completo esencial

- Actor Principal
Usuario
- Actores Involucrados e Intereses
Usuario: Es la persona que tiene baja visión.
- Precondiciones
 1. La aplicación ha sido ejecutada.
 2. El dispositivo móvil posee una cámara trasera.
 3. El dispositivo móvil tiene sistema operativo Android 4.0.
 4. El dispositivo tiene instalado OpenCV Manager.
- Garantías de Éxito (Post condiciones)
 1. El usuario se desplaza sobre una imagen fija.
 2. La aplicación desplaza la imagen fija en la dirección que desee el usuario.
- Escenario Principal de Éxito
 1. El usuario elige la opción “translate”.
- Extensiones o Flujos Alternativos
No se consideran.
- Temas Abiertos
No se consideran.
- Diagrama de Secuencia: figura 13

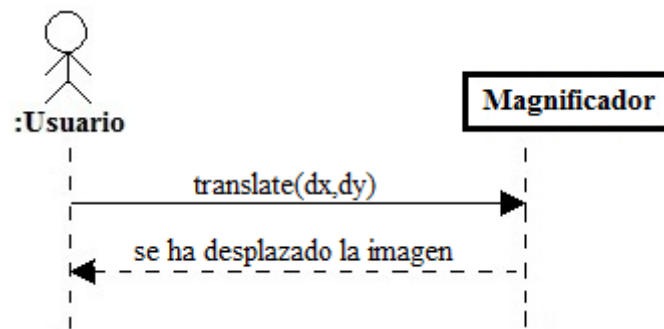


Figura 13: Diagrama de secuencia: Desplazarse sobre una imagen capturada.

4.2.2.9. Activar el flash

Descripción en formato breve

El usuario desea activar el flash del dispositivo para poder ver con más claridad.

Descripción en formato completo esencial

- Actor Principal
Usuario
- Actores Involucrados e Intereses
Usuario: Es la persona que tiene baja visión.
- Precondiciones
 1. La aplicación ha sido ejecutada.
 2. El dispositivo móvil posee una cámara trasera.
 3. El dispositivo móvil tiene flash.
 4. El dispositivo móvil tiene sistema operativo Android 4.0.
 5. El dispositivo tiene instalado OpenCV Manager.
- Garantías de Éxito (Post condiciones)
 1. La aplicación activa el flash.
- Escenario Principal de Éxito
 1. El usuario elige la opción “flashOn”.
- Extensiones o Flujos Alternativos
No se consideran.
- Temas Abiertos
No se consideran.
- Diagrama de Secuencia: figura 14

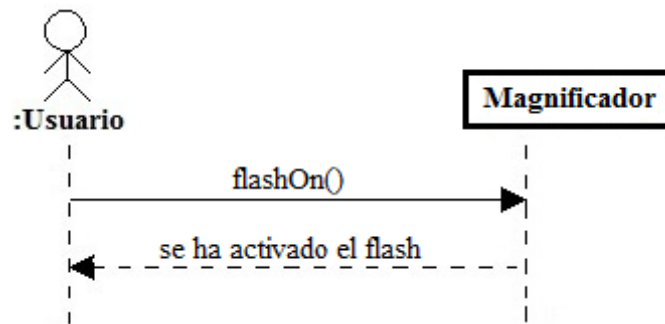


Figura 14: Diagrama de secuencia: Activar el flash.

4.2.2.10. Desactivar el flash

Descripción en formato breve

El usuario desea desactivar el flash del dispositivo.

Descripción en formato completo esencial

- Actor Principal
Usuario
- Actores Involucrados e Intereses
Usuario: Es la persona que tiene baja visión.
- Precondiciones
 1. La aplicación ha sido ejecutada.
 2. El dispositivo móvil posee una cámara trasera.
 3. El dispositivo móvil tiene flash.
 4. El dispositivo móvil tiene sistema operativo Android 4.0.
 5. El dispositivo tiene instalado OpenCV Manager.
- Garantías de Éxito (Post condiciones)
 1. La aplicación desactiva el flash.
- Escenario Principal de Éxito
 1. El usuario elige la opción “flashOff”.
- Extensiones o Flujos Alternativos
No se consideran.
- Temas Abiertos
No se consideran.
- Diagrama de Secuencia: figura 15

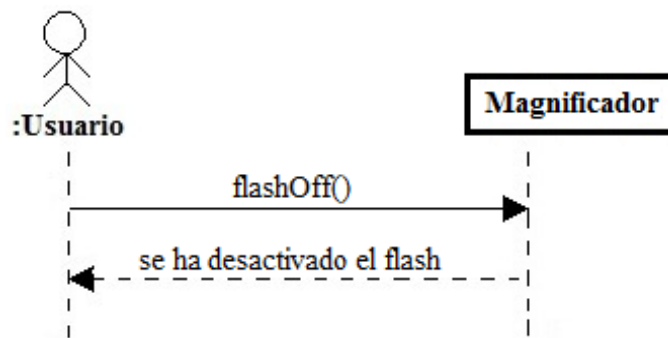


Figura 15: Diagrama de secuencia: Desactivar el flash.

4.2.2.11. Activar auto enfoque

Descripción en formato breve

El usuario desea activar el enfoque automático para que sea más nítido lo que observa.

Descripción en formato completo esencial

- Actor Principal
Usuario
- Actores Involucrados e Intereses
Usuario: Es la persona que tiene baja visión.
- Precondiciones
 1. La aplicación ha sido ejecutada.
 2. El dispositivo móvil posee una cámara trasera.
 3. El dispositivo móvil tiene sistema operativo Android 4.0.
 4. El dispositivo tiene instalado OpenCV Manager.
- Garantías de Éxito (Post condiciones)
 1. La aplicación activa el enfoque automático.
- Escenario Principal de Éxito
 1. El usuario elige la opción “autoFocus”.
- Extensiones o Flujos Alternativos
No se consideran.
- Temas Abiertos
No se consideran.
- Diagrama de Secuencia: figura 16

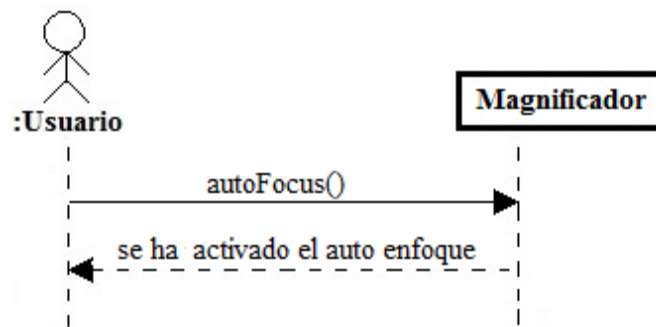


Figura 16: Diagrama de secuencia: Activar auto enfoque.

4.2.2.12. Desactivar auto enfoque

Descripción en formato breve

El usuario desea desactivar el enfoque automático del dispositivo móvil.

Descripción en formato completo esencial

- Actor Principal
Usuario
- Actores Involucrados e Intereses
Usuario: Es la persona que tiene baja visión.
- Precondiciones
 1. La aplicación ha sido ejecutada.
 2. El dispositivo móvil posee una cámara trasera.
 3. El dispositivo móvil tiene sistema operativo Android 4.0.
 4. El dispositivo tiene instalado OpenCV Manager.
- Garantías de Éxito (Post condiciones)
 1. La aplicación desactiva el enfoque automático.
- Escenario Principal de Éxito
 1. El usuario elige la opción “cancelAutoFocus”.
- Extensiones o Flujos Alternativos
No se consideran.
- Temas Abiertos
No se consideran.
- Diagrama de Secuencia: figura 17

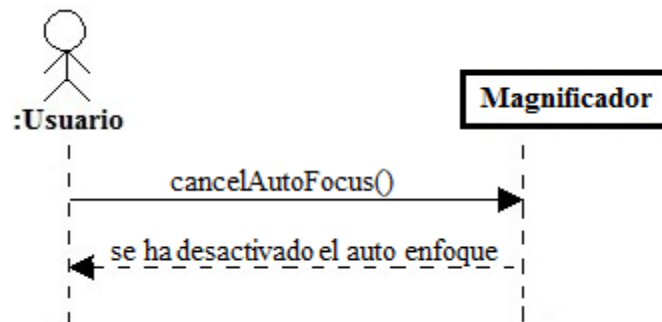


Figura 17: Diagrama de secuencia: Desactivar auto enfoque.

4.2.2.13. Activar macro enfoque

Descripción en formato breve

El usuario desea activar el macro enfoque del dispositivo móvil.

Descripción en formato completo esencial

- Actor Principal
Usuario
- Actores Involucrados e Intereses
Usuario: Es la persona que tiene baja visión.
- Precondiciones
 1. La aplicación ha sido ejecutada.
 2. El dispositivo móvil posee una cámara trasera.
 3. El dispositivo móvil tiene sistema operativo Android 4.0.
 4. El dispositivo tiene instalado OpenCV Manager.
- Garantías de Éxito (Post condiciones)
 1. La aplicación activa el macro enfoque.
- Escenario Principal de Éxito
 1. El usuario elige la opción “macroFocus”.
- Extensiones o Flujos Alternativos
No se consideran.
- Temas Abiertos
No se consideran.
- Diagrama de Secuencia: figura 18

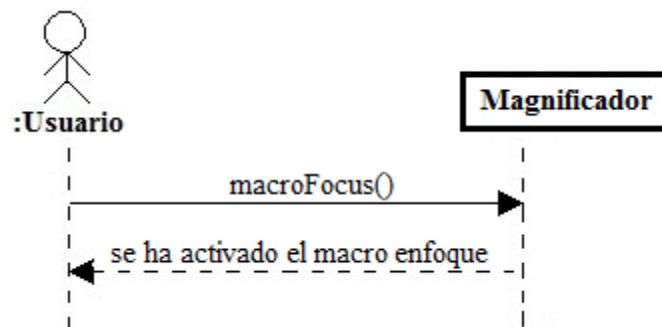


Figura 18: Diagrama de secuencia: Activar macro enfoque.

4.2.2.14. Activar enfoque de área

Descripción en formato breve

El usuario desea activar el enfoque de área del dispositivo móvil.

Descripción en formato completo esencial

- Actor Principal
Usuario
- Actores Involucrados e Intereses
Usuario: Es la persona que tiene baja visión.
- Precondiciones
 1. La aplicación ha sido ejecutada.
 2. El dispositivo móvil posee una cámara trasera.
 3. El dispositivo móvil tiene sistema operativo Android 4.0.
 4. El dispositivo tiene instalado OpenCV Manager.
- Garantías de Éxito (Post condiciones)
 1. La aplicación activa el enfoque de área.
- Escenario Principal de Éxito
 1. El usuario elige la opción “focusArea”.
- Extensiones o Flujos Alternativos
No se consideran.
- Temas Abiertos
No se consideran.
- Diagrama de Secuencia: figura 19

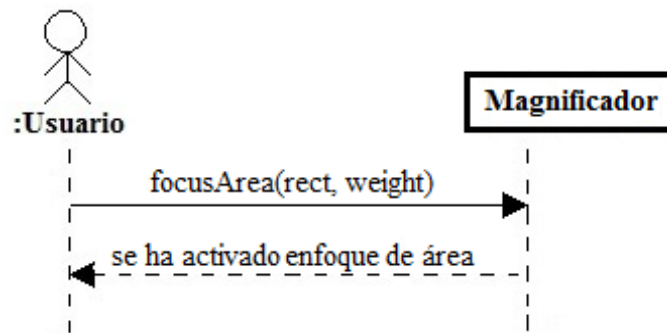


Figura 19: Diagrama de secuencia: Activar enfoque de área.

4.2.2.15. Activar enfoque de video

Descripción en formato breve

El usuario desea activar el enfoque de video del dispositivo móvil.

Descripción en formato completo esencial

- Actor Principal
Usuario
- Actores Involucrados e Intereses
Usuario: Es la persona que tiene baja visión.
- Precondiciones
 1. La aplicación ha sido ejecutada.
 2. El dispositivo móvil posee una cámara trasera.
 3. El dispositivo móvil tiene sistema operativo Android 4.0.
 4. El dispositivo tiene instalado OpenCV Manager.
- Garantías de Éxito (Post condiciones)
 1. La aplicación activa el enfoque de video.
- Escenario Principal de Éxito
 1. El usuario elige la opción “videoFocus”.
- Extensiones o Flujos Alternativos
No se consideran.
- Temas Abiertos
No se consideran.
- Diagrama de Secuencia: figura 20

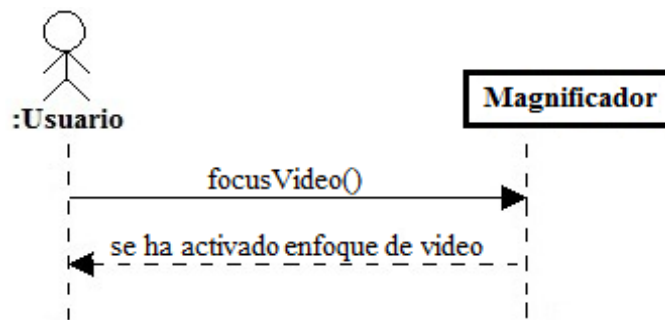


Figura 20: Diagrama de secuencia: Activar enfoque de video.

4.2.2.16. Activar low fps

Descripción en formato breve

El usuario desea activar la función de pocas imágenes por segundo del dispositivo móvil.

Descripción en formato completo esencial

- Actor Principal
Usuario
- Actores Involucrados e Intereses
Usuario: Es la persona que tiene baja visión.
- Precondiciones
 1. La aplicación ha sido ejecutada.
 2. El dispositivo móvil posee una cámara trasera.
 3. El dispositivo móvil tiene sistema operativo Android 4.0.
 4. El dispositivo tiene instalado OpenCV Manager.
- Garantías de Éxito (Post condiciones)
 1. La aplicación activa la función pocas imágenes por segundo.
- Escenario Principal de Éxito
 1. El usuario elige la opción “lowFps”.
- Extensiones o Flujos Alternativos
No se consideran.
- Temas Abiertos
No se consideran.
- Diagrama de Secuencia: figura 21

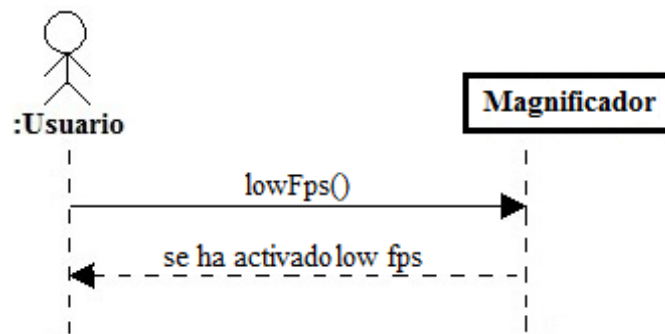


Figura 21: Diagrama de secuencia: Activar low fps.

4.2.2.17. Activar default fps

Descripción en formato breve

El usuario desea activar la función de imágenes por segundo por defecto del dispositivo móvil.

Descripción en formato completo esencial

- Actor Principal
Usuario
- Actores Involucrados e Intereses
Usuario: Es la persona que tiene baja visión.
- Precondiciones
 1. La aplicación ha sido ejecutada.
 2. El dispositivo móvil posee una cámara trasera.
 3. El dispositivo móvil tiene sistema operativo Android 4.0.
 4. El dispositivo tiene instalado OpenCV Manager.
- Garantías de Éxito (Post condiciones)
 1. La aplicación activa la función imágenes por segundo por defecto.
- Escenario Principal de Éxito
 1. El usuario elige la opción “defaultFps”.
- Extensiones o Flujos Alternativos
No se consideran.
- Temas Abiertos
No se consideran.
- Diagrama de Secuencia: figura 22

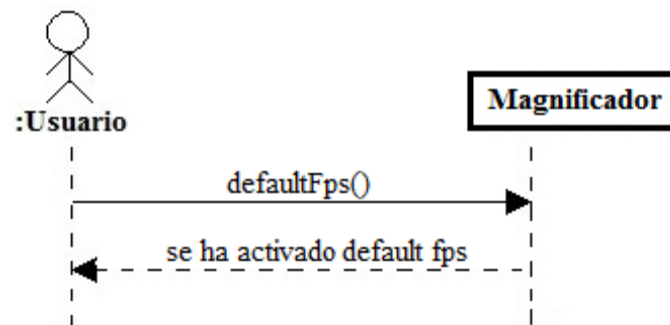


Figura 22: Diagrama de secuencia: Activar default fps.

4.2.2.18. Cambiar contraste

Descripción en formato breve

El usuario desea cambiar el contraste del dispositivo móvil.

Descripción en formato completo esencial

- Actor Principal
Usuario
- Actores Involucrados e Intereses
Usuario: Es la persona que tiene baja visión.
- Precondiciones
 1. La aplicación ha sido ejecutada.
 2. El dispositivo móvil posee una cámara trasera.
 3. El dispositivo móvil tiene sistema operativo Android 4.0.
 4. El dispositivo tiene instalado OpenCV Manager.
- Garantías de Éxito (Post condiciones)
 1. La aplicación cambia el contraste.
- Escenario Principal de Éxito
 1. El usuario elige la opción “contrast”.
- Extensiones o Flujos Alternativos
No se consideran.
- Temas Abiertos
No se consideran.
- Diagrama de Secuencia: figura 23

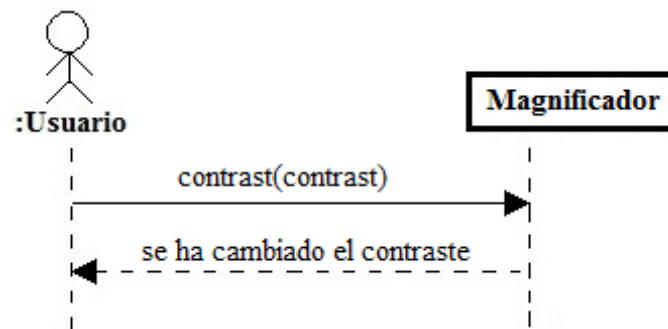


Figura 23: Diagrama de secuencia: Cambiar contraste.

4.2.2.19. Invertir colores

Descripción en formato breve

El usuario desea invertir los colores del dispositivo móvil.

Descripción en formato completo esencial

- Actor Principal
Usuario
- Actores Involucrados e Intereses
Usuario: Es la persona que tiene baja visión.
- Precondiciones
 1. La aplicación ha sido ejecutada.
 2. El dispositivo móvil posee una cámara trasera.
 3. El dispositivo móvil tiene sistema operativo Android 4.0.
 4. El dispositivo tiene instalado OpenCV Manager.
- Garantías de Éxito (Post condiciones)
 1. La aplicación invierte los colores.
- Escenario Principal de Éxito
 1. El usuario elige la opción “invert”.
- Extensiones o Flujos Alternativos
No se consideran.
- Temas Abiertos
No se consideran.
- Diagrama de Secuencia: figura 24

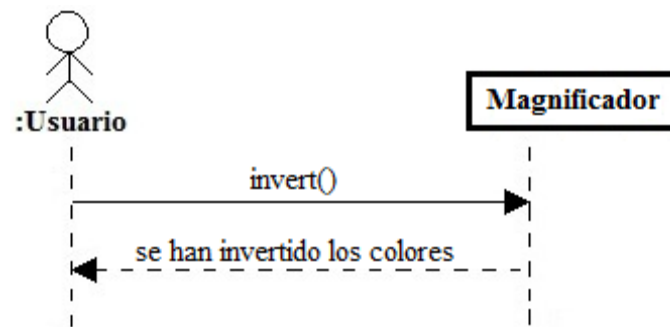


Figura 24: Diagrama de secuencia: Invertir colores.

4.2.2.20. Activar estabilización del video

Descripción en formato breve

El usuario desea activar la estabilización del video del dispositivo móvil.

Descripción en formato completo esencial

- Actor Principal
Usuario
- Actores Involucrados e Intereses
Usuario: Es la persona que tiene baja visión.
- Precondiciones
 1. La aplicación ha sido ejecutada.
 2. El dispositivo móvil posee una cámara trasera.
 3. El dispositivo móvil tiene sistema operativo Android 4.0.
 4. El dispositivo tiene instalado OpenCV Manager.
- Garantías de Éxito (Post condiciones)
 1. La aplicación activa la estabilización del video.
- Escenario Principal de Éxito
 1. El usuario elige la opción “videoStabilizationOn”.
- Extensiones o Flujos Alternativos
No se consideran.
- Temas Abiertos
No se consideran.
- Diagrama de Secuencia: figura 25

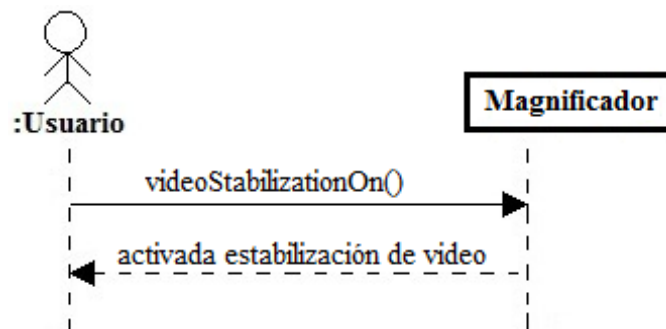


Figura 25: Diagrama de secuencia: Activar estabilización de video.

4.2.2.21. Desactivar estabilización del video

Descripción en formato breve

El usuario desea activar la estabilización del video del dispositivo móvil.

Descripción en formato completo esencial

- Actor Principal
Usuario
- Actores Involucrados e Intereses
Usuario: Es la persona que tiene baja visión.
- Precondiciones
 1. La aplicación ha sido ejecutada.
 2. El dispositivo móvil posee una cámara trasera.
 3. El dispositivo móvil tiene sistema operativo Android 4.0.
 4. El dispositivo tiene instalado OpenCV Manager.
- Garantías de Éxito (Post condiciones)
 1. La aplicación desactiva la estabilización del video.
- Escenario Principal de Éxito
 1. El usuario elige la opción “videoStabilizationOff”.
- Extensiones o Flujos Alternativos
No se consideran.
- Temas Abiertos
No se consideran.
- Diagrama de Secuencia: figura 26

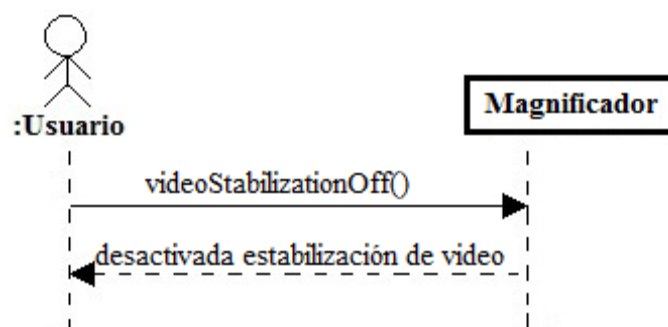


Figura 26: Diagrama de secuencia: Desactivar estabilización de video.

4.3. Implementación y pruebas

Para la implementación, se eligió el sistema operativo Ubuntu v10.0 y para el desarrollo del código en sí, se utilizó el entorno de programación Eclipse Juno. Para poder tener acceso a las funciones propias de Android fue necesario instalar la extensión Android SDK a Eclipse.

El proyecto que se creó en Eclipse fue para la versión de Android 4.1.2 con la API 16. En cuanto al procesamiento de la imagen se utilizó la librería OpenCV.

Dada la escasa experiencia en el tema, la implementación a bajo nivel siguió una filosofía de prueba-error. Cada modificación en el código era necesaria probarla directamente en el dispositivo para comprobar si funciona o daba error. Si se encontraba algún problema, se consultaba con diferentes tutoriales, guías, ejemplos o foros. Para llevar a cabo la realización de las pruebas, se decidió dividir éstas en tres bloques: pruebas de módulo cámara, pruebas de módulo librería y pruebas de sistema. Se han hecho varias iteraciones de pruebas ya que las primeras no daban resultados deseados. Era necesario depurar el código, modificar, cambiar el algoritmo, etc.

Para tener acceso a las funciones de la librería del magnificador, se ha hecho uso del menú genérico de las aplicaciones Android, ya que al no disponer de la interfaz de usuario porque no es el objetivo de este Trabajo de Fin de Grado, ésta era la más simple de implementar para poder realizar las pruebas. En algunas de las funciones que requerían el paso de parámetros, por ejemplo el aumentar la imagen, se utilizaron valores estáticos o con incrementos de $x+0.2$ cada vez que se llamase a la función, ya que el menú genérico no permitía otro modo de funcionamiento.

La implementación de las pruebas se encuentra en el fichero `MagnificadorActivity.java`. Para las pruebas de sistema fue necesario instalar la aplicación previamente compilada en los dispositivos probados.

4.3.1. Pruebas de módulo cámara

Información de las Pruebas de Modulo: Cámara

ID Prueba	Pruebas_Camara1	Iteración	1.0
Fecha Prueba	21/11/2012	Dispositivo	Sony Xperia U Android v4.0

Módulos probados

Cámara (parte del fichero MagnificadorBase.java y MagnificadorProcess.java)

Descripción de la prueba

Esta prueba incluye: ejecutar la aplicación en el dispositivo Sony Xperia U, cargar la librería OpenCV, inicializar la cámara y video en RGB.

Subprueba	Resultado Esperado	Resultado Obtenido	Causa/Solución
Ejecutar App	Sin error	NoClassDefFoundError	Error en el fichero manifest.xml
Carga OpenCV lib	Sin error	-	La aplicación no ha sido ejecutada por el error anterior
Inicialización cámara	Sin error	-	La aplicación no ha sido ejecutada por el error anterior
RGB	Sin error	-	La aplicación no ha sido ejecutada por el error anterior

ID Prueba	Pruebas_Camara2	Iteración	2.0
Fecha Prueba	21/11/2012	Dispositivo	Sony Xperia U Android v4.0

Módulos probados

Cámara (parte del fichero MagnificadorBase.java y MagnificadorProcess.java)

Descripción de la prueba

Esta prueba incluye: ejecutar la aplicación en el dispositivo Sony Xperia U, cargar la librería OpenCV, inicializar la cámara y video en RGB.

Subprueba	Resultado Esperado	Resultado Obtenido	Causa/Solución
Ejecutar App	Sin error	Sin error	-

Carga OpenCV lib	Sin error	Sin error	-
Inicialización cámara	Sin error	NullPointerException	Parámetros incorrectos para inicializar la camera.
RGB	Sin error	-	La aplicación no ha sido ejecutada por el error anterior

ID Prueba	Pruebas_Camara3	Iteración	3.0
Fecha Prueba	21/11/2012	Dispositivo	Sony Xperia U Android v4.0

Módulos probados

Cámara (parte del fichero MagnificadorBase.java y MagnificadorProcess.java)

Descripción de la prueba

Esta prueba incluye: ejecutar la aplicación en el dispositivo Sony Xperia U, cargar la librería OpenCV, inicializar la cámara y video en RGB.

Subprueba	Resultado Esperado	Resultado Obtenido	Causa/Solución
Ejecutar App	Sin error	Sin error	-
Carga OpenCV lib	Sin error	Sin error	-
Inicialización cámara	Sin error	Sin error	-
RGB	Sin error	Vista en gris	Error en la conversión de la imagen - uso incorrecto de la librería OpenCV.

ID Prueba	Pruebas_Camara4	Iteración	4.0
Fecha Prueba	21/11/2012	Dispositivo	Sony Xperia U Android v4.0

Módulos probados

Cámara (parte del fichero MagnificadorBase.java y MagnificadorProcess.java)

Descripción de la prueba

Esta prueba incluye: ejecutar la aplicación en el dispositivo Sony Xperia U, cargar la librería OpenCV, inicializar la cámara y video en RGB.

Subprueba	Resultado Esperado	Resultado Obtenido	Causa/Solución
Ejecutar App	Sin error	Sin error	-
Carga OpenCV lib	Sin error	Sin error	-
Inicialización cámara	Sin error	Sin error	-
RGB	Sin error	Sin error	-

4.3.2. Pruebas de módulo librería

Información de las Pruebas de Modulo: Librería magnificador

ID Prueba	Pruebas_Librería1	Iteración	1.0
Fecha Prueba	22/12/2012	Dispositivo	Sony Xperia U Android v4.0

Módulos probados

Librería magnificador (parte del fichero MagnificadorBase.java y MagnificadorProcess.java)

Descripción de la prueba

Esta prueba incluye: probar todas las funciones de la librería magnificador usando para esto el menú genérico de Android. Este módulo depende del módulo de la cámara por lo tanto fue necesario que este último funcionase correctamente.

Subprueba	Resultado Esperado	Resultado Obtenido	Comentario
Video: Cambiar de RGB a Gray	Sin error- Vista en escala de grises	Sin error-Vista en escala de grises	-
Video: Cambiar de RGB a BW	Sin error- Vista e blanco y negro	Funcionamiento incorrecto- Pantalla en negro	Valores incorrectos de los parámetros usados para la conversión de la imagen.
Video: Cambiar de RGB a BGR	Sin error- Vista en modo BGR	Sin error-Vista en modo BGR	-
Video: Cambiar de Gray a RGB	Sin error- Vista en modo RGB	Sin error-Vista en modo RGB	-
Video: Cambiar de Gray a BW	Sin error- Vista e blanco y negro	Funcionamiento incorrecto- Pantalla en negro	Valores incorrectos de los parámetros usados para la conversión de la imagen.
Video: Cambiar de Gray a BGR	Sin error- Vista en modo BGR	Sin error-Vista en modo BGR	-

Video: Cambiar de BW a RGB	Sin error- Vista en modo RGB	Sin error-Vista en modo RGB	En modo BW la pantalla está en negro
Video: Cambiar de BW a Gray	Sin error- Vista en escala de grises	Sin error-Vista en escala de grises	En modo BW la pantalla está en negro
Video: Cambiar de BW a BGR	Sin error- Vista en modo BGR	Sin error-Vista en modo BGR	En modo BW la pantalla está en negro
Video: Cambiar de BGR a RGB	Sin error- Vista en modo RGB	Sin error-Vista en modo RGB	-
Video: Cambiar de BGR a Gray	Sin error- Vista en escala de grises	Sin error-Vista en escala de grises	-
Video: Cambiar de BGR a BW	Sin error- Vista e blanco y negro	Funcionamiento incorrecto- Pantalla en negro	Valores incorrectos de los parámetros usados para la conversión de la imagen.
Video: Invertir colores	Sin error- Colores invertidos	Sin error- Colores invertidos	-
Video: Estabilización de imagen	No se ha probado	No se ha probado	No se disponía del móvil capaz de soportar estabilización de video. Algoritmo propio de Android, se considera que funciona sin problemas.
Video: Modificar contraste	Sin error – Contraste modificado	Sin error – Contraste modificado	-
Video: Aumentar hasta 5 veces.	Sin error- Video aumentado	Sin error- Video aumentado	Para esta prueba se usó el aumento usando saltos discretos.
Video: Disminuir hasta vista sin aumento	Sin error- Video sin aumento	Sin error- Video sin aumento	Para esta prueba se usó el aumento usando saltos discretos.
Video: Disminuir vista sin aumento	Sin error- Vista sobrepuesta-doble	Sin error- Vista sobrepuesta-doble	Los parámetros que se pasa a la función que hace el aumento/disminución no tienen ningún límite por lo tanto pueden ofrecer comportamiento anómalo aunque previsto. Estos valores deberían controlarse en el momento de llamar al método.
Video: Activar flash	Sin error- Flash activado	Sin error- Flash activado	-
Video: Desactivar flash	Sin error- Flash desactivado	Sin error- Flash desactivado	-
Video: Activar enfoque video	Sin error- Enfoque video activo	Sin error- Enfoque video activo	-
Video: Activar autoenfoque	Sin error- Autoenfoque activo	Sin error- Autoenfoque activo	-
Video: Cancelar autoenfoque	Sin error- Autoenfoque desactivado	Sin error- Autoenfoque desactivado	-
Video: Activar enfoque macro	Sin error- Enfoque macro activo	Sin error- Enfoque macro activo	-

Video: Activar enfoque área	Sin error- Enfoque en área activo	Sin error- Enfoque en área activo	Para probar esta funcionalidad fue necesario usar valores constantes ya que la interacción por el menú genérico no ofrece otra posibilidad.
Video: FPS bajo	Sin error- Refresco de la imagen más lento	Sin error- Refresco de la imagen más lento	-
Video: FPS normal	Sin error- Refresco de la imagen estándar	Sin error- Refresco de la imagen estándar	-
Video: Pausar el video	Sin error- Video congelado	Sin error- Video congelado	-
Imagen: Cambiar de RGB a Gray	Sin error- Vista en escala de grises	Funcionamiento incorrecto- Vista en escala de grises pero no de la imagen capturada	La vista pasa a escala de grises pero no sobre la imagen capturada sino sobre una nueva imagen a la que está apuntando actualmente la cámara.
Imagen: Cambiar de RGB a BW	Sin error- Vista en blanco y negro	Funcionamiento incorrecto- Pantalla en negro	Valores incorrectos de los parámetros usados para la conversión de la imagen.
Imagen: Cambiar de RGB a BGR	Sin error- Vista en modo BGR	Funcionamiento incorrecto- Vista en BGR pero no de la imagen capturada	La vista pasa a BGR pero no sobre la imagen capturada sino sobre una nueva imagen a la que está apuntando actualmente la cámara.
Imagen: Cambiar de Gray a RGB	Sin error- Vista en modo RGB	Funcionamiento incorrecto- Vista en RGB pero no de la imagen capturada	La vista pasa a RGB pero no sobre la imagen capturada sino sobre una nueva imagen a la que está apuntando actualmente la cámara.
Imagen: Cambiar de Gray a BW	Sin error- Vista en blanco y negro	Pantalla en negro	Valores incorrectos de los parámetros usados para la conversión de la imagen.
Imagen: Cambiar de Gray a BGR	Sin error- Vista en modo BGR	Funcionamiento incorrecto- Vista en BGR pero no de la imagen capturada	La vista pasa a BGR pero no sobre la imagen capturada sino sobre una nueva imagen a la que está apuntando actualmente la cámara.
Imagen: Cambiar de BW a RGB	Sin error- Vista en modo RGB	Funcionamiento incorrecto- Vista en RGB pero no de la imagen capturada	En modo BW la pantalla está en negro. La vista pasa a RGB pero no sobre la imagen capturada sino sobre una nueva imagen a la que está apuntando actualmente la cámara.
Imagen: Cambiar de BW a Gray	Sin error- Vista en escala de grises	Funcionamiento incorrecto- Vista en	En modo BW la pantalla está en negro. La vista

		escala de grises pero no de la imagen capturada	pasa a escala de grises pero no sobre la imagen capturada sino sobre una nueva imagen a la que está apuntando actualmente la cámara.
Imagen: Cambiar de BW a BGR	Sin error- Vista en modo BGR	Funcionamiento incorrecto- Vista en BGR pero no de la imagen capturada	En modo BW la pantalla está en negro. La vista pasa a BGR pero no sobre la imagen capturada sino sobre una nueva imagen a la que está apuntando actualmente la cámara.
Imagen: Cambiar de BGR a RGB	Sin error- Vista en modo RGB	Funcionamiento incorrecto- Vista en RGB pero no de la imagen capturada	La vista pasa a RGB pero no sobre la imagen capturada sino sobre una nueva imagen a la que está apuntando actualmente la cámara.
Imagen: Cambiar de BGR a Gray	Sin error- Vista en escala de grises	Funcionamiento incorrecto- Vista en escala de grises pero no de la imagen capturada	La vista pasa a escala de grises pero no sobre la imagen capturada sino sobre una nueva imagen a la que está apuntando actualmente la cámara.
Imagen: Cambiar de BGR a BW	Sin error- Vista e blanco y negro	Pantalla en negro	Valores incorrectos de los parámetros usados para la conversión de la imagen.
Imagen: Invertir colores	Sin error- Colores invertidos	Sin error- Colores invertidos	-
Imagen: Modificar contraste	Sin error – Contraste modificado	Sin error – Contraste modificado	-
Imagen: Aumentar hasta 5 veces.	Sin error- Imagen aumentado	Sin error- Imagen aumentado	Para esta prueba se usó el aumento usando saltos discretos.
Imagen: Disminuir hasta vista sin aumento	Sin error- Imagen sin aumento	Sin error- Imagen sin aumento	Para esta prueba se usó el aumento usando saltos discretos.
Video: Disminuir vista sin aumento	Sin error- Vista sobrepuesta-doble	Sin error- Vista sobrepuesta-doble	Los parámetros que se pasa a la función que hace el aumento/disminución no tienen ningún límite por lo tanto pueden ofrecer comportamiento anómalo aunque previsto. Estos valores deberían controlarse en el momento de llamar al método.
Imagen: Trasladar hacia arriba	Sin error – Imagen trasladada	Sin error – Imagen trasladada	El traslado de la imagen se hizo usando valores discretos.

Imagen: Trasladar hacia abajo	Sin error – Imagen trasladada	Sin error – Imagen trasladada	El traslado de la imagen se hizo usando valores discretos.
Imagen: Trasladar hacia derecha	Sin error – Imagen trasladada	Sin error – Imagen trasladada	El traslado de la imagen se hizo usando valores discretos.
Imagen: Trasladar hacia izquierda	Sin error – Imagen trasladada	Sin error – Imagen trasladada	El traslado de la imagen se hizo usando valores discretos.
Imagen: Trasladar imagen pasándose de sus bordes	Sin error- Vista sobrepuesta-doble	Sin error- Vista sobrepuesta-doble	Los parámetros que se pasa a la función que hace el traslado no tienen ningún límite por lo tanto pueden ofrecer comportamiento anómalo aunque previsto. Estos valores deberían controlarse en el momento de llamar al método.
Imagen: Activar el video	Sin error- Video	Sin error- Video	-

ID Prueba	Pruebas_Librería2	Iteración	2.0
Fecha Prueba	22/12/2012	Dispositivo	Sony Xperia U Android v4.0

Módulos probados

Librería magnificador (parte del fichero MagnificadorBase.java y MagnificadorProcess.java)

Descripción de la prueba

Esta prueba incluye: probar todas las funciones de la librería magnificador usando para esto el menú genérico de Android. Este módulo depende del módulo de la cámara por lo tanto fue necesario que este último funcionase correctamente.

Subprueba	Resultado Esperado	Resultado Obtenido	Comentario
Video: Cambiar de RGB a Gray	Sin error- Vista en escala de grises	Sin error-Vista en escala de grises	-
Video: Cambiar de RGB a BW	Sin error- Vista e blanco y negro	Sin error- Vista en blanco y negro	-
Video: Cambiar de RGB a BGR	Sin error- Vista en modo BGR	Sin error-Vista en modo BGR	-
Video: Cambiar de Gray a RGB	Sin error- Vista en modo RGB	Sin error-Vista en modo RGB	-
Video: Cambiar de Gray a BW	Sin error- Vista e blanco y negro	Sin error- Vista en blanco y negro	-
Video: Cambiar de Gray a BGR	Sin error- Vista en modo BGR	Sin error-Vista en modo BGR	-

Video: Cambiar de BW a RGB	Sin error- Vista en modo RGB	Sin error-Vista en modo RGB	-
Video: Cambiar de BW a Gray	Sin error- Vista en escala de grises	Sin error-Vista en escala de grises	-
Video: Cambiar de BW a BGR	Sin error- Vista en modo BGR	Sin error-Vista en modo BGR	-
Video: Cambiar de BGR a RGB	Sin error- Vista en modo RGB	Sin error-Vista en modo RGB	-
Video: Cambiar de BGR a Gray	Sin error- Vista en escala de grises	Sin error-Vista en escala de grises	-
Video: Cambiar de BGR a BW	Sin error- Vista e blanco y negro	Sin error- Vista en blanco y negro	-
Video: Invertir colores	Sin error- Colores invertidos	Sin error- Colores invertidos	-
Video: Estabilización de imagen	No se ha probado	No se ha probado	No se disponía del móvil capaz de soportar estabilización de video. Algoritmo propio de Android, se considera que funciona sin problemas.
Video: Modificar contraste	Sin error – Contraste modificado	Sin error – Contraste modificado	-
Video: Aumentar hasta 5 veces.	Sin error- Video aumentado	Sin error- Video aumentado	Para esta prueba se usó el aumento usando saltos discretos.
Video: Disminuir hasta vista sin aumento	Sin error- Video sin aumento	Sin error- Video sin aumento	Para esta prueba se usó el aumento usando saltos discretos.
Video: Disminuir vista sin aumento	Sin error- Vista sobrepuesta-doble	Sin error- Vista sobrepuesta-doble	Los parámetros que se pasa a la función que hace el aumento/disminución no tienen ningún límite por lo tanto pueden ofrecer comportamiento anómalo aunque previsto. Estos valores deberían controlarse en el momento de llamar al método.
Video: Activar flash	Sin error- Flash activado	Sin error- Flash activado	-
Video: Desactivar flash	Sin error- Flash desactivado	Sin error- Flash desactivado	-
Video: Activar enfoque video	Sin error- Enfoque video activo	Sin error- Enfoque video activo	-
Video: Activar autoenfoque	Sin error- Autoenfoque activo	Sin error- Autoenfoque activo	-
Video: Cancelar autoenfoque	Sin error- Autoenfoque desactivado	Sin error- Autoenfoque desactivado	-
Video: Activar enfoque macro	Sin error- Enfoque macro activo	Sin error- Enfoque macro activo	-
Video: Activar enfoque área	Sin error- Enfoque en área activo	Sin error- Enfoque en área activo	Para probar esta funcionalidad fue

			necesario usar valores constantes ya que la interacción por el menú genérico no ofrece otra posibilidad.
Video: FPS bajo	Sin error- Refresco de la imagen más lento	Sin error- Refresco de la imagen más lento	-
Video: FPS normal	Sin error- Refresco de la imagen estándar	Sin error- Refresco de la imagen estándar	-
Video: Pausar el video	Sin error- Imagen fija	Sin error- Imagen fija	-
Imagen: Cambiar de RGB a Gray	Sin error- Vista en escala de grises	Sin error- Vista en escala de grises	-
Imagen: Cambiar de RGB a BW	Sin error- Vista en blanco y negro	Sin error- Vista en blanco y negro	-
Imagen: Cambiar de RGB a BGR	Sin error- Vista en modo BGR	Sin error- Vista en modo BGR	-
Imagen: Cambiar de Gray a RGB	Sin error- Vista en modo RGB	Sin error- Vista en modo RGB	-
Imagen: Cambiar de Gray a BW	Sin error- Vista en blanco y negro	Sin error- Vista en blanco y negro	-
Imagen: Cambiar de Gray a BGR	Sin error- Vista en modo BGR	Sin error- Vista en modo BGR	-
Imagen: Cambiar de BW a RGB	Sin error- Vista en modo RGB	Sin error- Vista en modo RGB	-
Imagen: Cambiar de BW a Gray	Sin error- Vista en escala de grises	Sin error- Vista en escala de grises	-
Imagen: Cambiar de BW a BGR	Sin error- Vista en modo BGR	Sin error- Vista en modo BGR	-
Imagen: Cambiar de BGR a RGB	Sin error- Vista en modo RGB	Sin error- Vista en modo RGB	-
Imagen: Cambiar de BGR a Gray	Sin error- Vista en escala de grises	Sin error- Vista en escala de grises	-
Imagen: Cambiar de BGR a BW	Sin error- Vista e blanco y negro	Sin error- Vista e blanco y negro	-
Imagen: Invertir colores	Sin error- Colores invertidos	Sin error- Colores invertidos	-
Imagen: Modificar contraste	Sin error – Contraste modificado	Sin error – Contraste modificado	-
Imagen: Aumentar hasta 5 veces.	Sin error- Imagen aumentado	Sin error- Imagen aumentado	Para esta prueba se usó el aumento usando saltos discretos.
Imagen: Disminuir hasta vista sin aumento	Sin error- Imagen sin aumento	Sin error- Imagen sin aumento	Para esta prueba se usó el aumento usando saltos discretos.
Imagen: Trasladar hacia arriba	Sin error – Imagen trasladada	Sin error – Imagen trasladada	El traslado de la imagen se hizo usando valores discretos.
Imagen: Trasladar hacia abajo	Sin error – Imagen trasladada	Sin error – Imagen trasladada	El traslado de la imagen se hizo usando valores discretos.
Imagen: Trasladar hacia derecha	Sin error – Imagen trasladada	Sin error – Imagen trasladada	El traslado de la imagen se hizo usando valores discretos.

Imagen: Trasladar hacia izquierda	Sin error – Imagen trasladada	Sin error – Imagen trasladada	El traslado de la imagen se hizo usando valores discretos.
Imagen: Activar el video	Sin error- Video	Sin error- Video	-

4.3.3. Pruebas de sistema

Información de las Pruebas de Sistema

ID Prueba	Pruebas_Sistema1	Iteración	1.0
Fecha Prueba	28/12/2012	Dispositivo	Sony Xperia S Android v4.0.4

Módulos probados

Módulo de cámara y librería magnificador conjuntas.

Descripción de la prueba

Esta prueba es similar a la prueba de módulo de la Liberia, pero se efectuó después de conseguir que todas las subpruebas en las pruebas de módulo funcionasen correctamente. Se usó un dispositivo diferente del que se usó en la implementación y pruebas de modulo.

Subprueba	Resultado Esperado	Resultado Obtenido	Comentario
Ejecutar la aplicación sin el OpenCV Manager	Sin error- Notificar que no se encontró el OpenCv Manager y preguntar si instalarlo.	Sin error- Notificar que no se encontró el OpenCv Manager y preguntar si instalarlo.	Al aceptar lleva a Play Market con el OpenCV Manager para instalarlo, se aceptó la descarga y la instalación.
Ejecutar la aplicación con el OpenCV Manager	Sin error- Ejecutar la aplicación y vista en RGB	Sin error- Ejecutar la aplicación y vista en RGB	-
Repetir las subpruebas de las pruebas de módulo librería	Sin error- resultados iguales a los de la 2ª iteración de pruebas de modulo librería.	Sin error- Ejecutar la aplicación y vista en RGB	-

ID Prueba	Pruebas_Sistema2	Iteración	1.0
Fecha Prueba	29/12/2012	Dispositivo	LG L3 Optimus Android v4.0

Módulos probados

Módulo de cámara y librería magnificador conjuntas.

Descripción de la prueba

Esta prueba es similar a la prueba de módulo de la Liberia, pero se efectuó después de conseguir que todas las subpruebas en las pruebas de módulo funcionasen correctamente. Se usó un dispositivo diferente del que se usó en la implementación y pruebas de modulo.

Subprueba	Resultado Esperado	Resultado Obtenido	Comentario
Ejecutar la aplicación sin el OpenCV Manager	Sin error- Notificar que no se encontró el OpenCv Manager y preguntar si instalarlo.	Sin error- Notificar que no se encontró el OpenCv Manager y preguntar si instalarlo.	Al aceptar nos lleva a Play Market con el OpenCV Manager para instalarlo, se aceptó la descarga y la instalación.
Ejecutar la aplicación con el OpenCV Manager	Sin error- Ejecutar la aplicación y vista en RGB	Sin error- Ejecutar la aplicación y vista en RGB	-
Repetir las subpruebas de las pruebas de módulo librería	Sin error- resultados iguales a los de la 2º iteración de pruebas de modulo librería.	Sin error- Ejecutar la aplicación y vista en RGB	-

ID Prueba	Pruebas_Sistema3	Iteración	1.0
Fecha Prueba	29/12/2012	Dispositivo	SE Xperia Arc Android v2.0

Módulos probados

Módulo de cámara y librería magnificador conjuntas.

Descripción de la prueba

Esta prueba es similar a la prueba de módulo de la Liberia, pero se efectuó después de conseguir que todas las subpruebas en las pruebas de módulo funcionasen correctamente. Se usó un dispositivo diferente del que se usó en la implementación y pruebas de módulo.

Subprueba	Resultado Esperado	Resultado Obtenido	Comentario
Ejecutar la aplicación sin el OpenCV Manager	Sin error- Notificar que no se encontró el OpenCv Manager y preguntar si instalarlo.	Sin error- Notificar que no se encontró el OpenCv Manager y preguntar si instalarlo.	Al aceptar lleva a Play Market con el OpenCV Manager para instalarlo, se aceptó la descarga y la instalación.
Ejecutar la aplicación con el OpenCV Manager	Sin error- Ejecutar la aplicación y vista en RGB	Sin error- Ejecutar la aplicación y vista en RGB	-
Repetir las subpruebas de las pruebas de módulo librería	Sin error- resultados iguales a los de la 2º iteración de pruebas de modulo librería.	Sin error- Ejecutar la aplicación y vista en RGB	-

5. Resultados y Conclusiones

Los resultados obtenidos después de realizar las pruebas fueron muy positivos. A pesar de que las primeras iteraciones de las pruebas presentaron algunos errores, éstos se eliminaron sin mayor problema.

La aplicación ha sido probada en varios dispositivos móviles, todos con Android 4.0. Puede parecer que la necesidad de instalar el programa adicional OpenCV Manager, sea un inconveniente pero los propios autores de la librería OpenCV insisten en que dicho programa debería estar instalado en el dispositivo, independientemente de las aplicaciones que lo utilicen. De esta manera, cualquier aplicación que use la librería OpenCV no tendrá que tenerla incorporada sino que utilizará la que esté instalada en el sistema. Así se garantiza que la versión de la librería es la más actual (actualizaciones automáticas) y se ahorra el espacio en la memoria del dispositivo.

También hay que señalar que aunque la aplicación ha funcionado en todos los dispositivos que se utilizaron para realizar las pruebas de sistema, el mejor comportamiento ha sido en los dispositivos con un procesador más potente; las conversiones de las imágenes son demasiado costosas para los aparatos más antiguos. Aunque es una clara desventaja, la tecnología móvil es muy dinámica y avanza rápidamente cada vez ofreciendo dispositivos más poderosos. También hay que tener en cuenta que mientras la interfaz de la aplicación esté lista, la mayoría de los usuarios tendrán aparatos que puedan soportar la aplicación sin ningún esfuerzo.

Es importante destacar que uno de los objetivos que se tenían pensados no fue implementado: vista en dos colores seleccionables. La razón por la que no se implementó la vista en dos colores seleccionables fue la falta de tiempo.

Con excepción de esta función, se implementaron todos los demás acordados con el tutor. Se ha intentado que el uso de la librería fuese lo más fácil posible para la persona que tenga que implementar la interfaz. Hay que destacar que la librería es independiente de la interacción que se va a ofrecer en el futuro, permitiendo una investigación que identifique la interfaz más usable para los futuros usuarios, teniendo en cuenta que son personas con discapacidad visual. No obstante, la persona que se dedique a esta tarea debería tener un conocimiento y experiencia mínima en programación Android y Java. De todos modos, se ha ofrecido un manual de usuario de la librería que se puede encontrar en el apartado Anexos.

El diseño final de la aplicación se ha modificado varias veces. Al principio, no se pensó que sería necesario un *thread* adicional que haga el procesamiento en tiempo real de la imagen. Pero los experimentos concluyeron que para ofrecerlo es imprescindible crear un *thread* que lo haga. De tal manera que aunque se use el término librería, el trabajo que se ha realizado es más que una librería común, que se incluye en un proyecto

usualmente. Este Trabajo incluye la base de una aplicación. Realmente, es una aplicación, inacabada por falta de una buena interfaz de usuario que está previsto desarrollar en otro Trabajo de Fin de Grado. Cuando ésta se incorpore en el proyecto, la aplicación estará terminada, lista para ofrecer sus funciones a las personas que lo necesiten.

Concluyendo, después de conocer los detalles de este proyecto, el éxito de la aplicación va a depender en su mayoría de la interfaz que se va a implementar. El núcleo de la aplicación funciona correctamente como se puede observar en los resultados de las pruebas que fueron realizadas, aunque la aplicación debería testearse en muchos más dispositivos antes de salir al mercado una vez esté lista.

Finalmente, hay que remarcar que el trabajo que se efectuó cumplió con los objetivos establecidos, los directos – crear la base de una aplicación de magnificación - y los indirectos, personales – aprender principios básicos de la programación para Android y profundizar los conocimientos adquiridos anteriormente en el campo de visión por ordenador. Éstos eran los objetivos a la hora de elegir este Trabajo Fin de Grado. Se considera que han sido cumplidos sumamente.

6. Futuras líneas de trabajo

En este apartado se van a identificar las principales líneas de trabajo que deberían tenerse en cuenta para el futuro.

El tema más importante, es dotar a la aplicación una interacción usable. El código en el estado actual presenta muchas maneras de implementarla. La implementación de la interfaz puede hacerse en una clase nueva, o también en la clase `MagnificadorActivity.java`. Para acceder a las funciones de la librería hay que crear un objeto de tipo `MagnificadorProcess`. Por ejemplo en el caso de una interfaz de gestos se puede usar funciones: `onTouchEvent(MotionEvent event)` y `onScale(ScaleGestureDetector detector)`. En la primera se implementan los casos de actividad (toques en la pantalla) que se pueden hacer en la aplicación, mientras la segunda función sirve para recibir un valor que identifica la cantidad a magnificar. Es recomendable usar todos los servicios que ofrece Android según la interfaz que se va a ofrecer, ya que se ahorrará el trabajo y el modo de interacción será más eficaz.

Como ya se hizo mención anteriormente, uno de los objetivos que se tenían pensados no fue implementado: vista en dos colores seleccionables. La razón por la que no se implementó la vista en dos colores seleccionables fue la falta de tiempo. La formación en las tecnologías desconocidas pero necesarias para la finalización del trabajo llevó más tiempo del previsto. El algoritmo que permitiría seleccionar dos colores no parece ser demasiado complejo (obtener imagen en blanco y negro, y sustituirlos por los colores deseados) pero las pruebas de su implementación no dieron resultados positivos. En su lugar se tomó la decisión de ofrecer la vista en BGR, que también puede ser muy útil para las personas. La implementación de vista en colores seleccionables queda pendiente como las posibles futuras líneas de trabajo.

También hay que decir que aunque se ha hablado solo de vista de colores seleccionables, el estado actual de la aplicación está preparado para la incorporación cualquier cantidad de funcionalidades, sin realizar mayores cambios al código existente – solamente será necesario añadir un caso más en la función que procesa la imagen (clase `MagnificadorProcess.java`, método `processFrame`), incorporar su algoritmo devolviendo una *bitmap* de puntos que forman la imagen procesada. Una vez tenido un nuevo caso de procesamiento de la imagen, hay que añadir una función en la clase `MagnificadorBase.java` que modifique la variable “*option*” asignándole el valor del caso en el cual se encuentra el nuevo algoritmo.

Es muy probable que, en un futuro las cámaras de los móviles ofrezcan un zoom óptico como una característica estándar. Para ese entonces sería muy recomendable hacer uso de ello, ya que la magnificación de ese tipo ofrece una calidad mucho mejor que cualquier digital, que es la que se utilizó en este Trabajo. Esta mejora, afectaría en un

grado mayor a la estructura básica de la aplicación, por lo cual no es recomendable incorporar una magnificación de dicho tipo en el proyecto actual, pero si es lo que se desea, lo más óptimo será crear una nueva aplicación donde el núcleo sea la magnificación óptica en lugar de la digital. En tal caso, este proyecto puede servir perfectamente como una guía y ejemplo para estudiar la programación de Android con el uso de la cámara y aprovechamiento de la librería OpenCV.

Otro factor importante, es que la estabilización de la imagen que se implementó es la que ofrece el propio Android en la versión 15 de API. Es necesario un dispositivo que ofrezca este servicio.

7. Bibliografía

- [3pies, 2012] 3pies (2012, Oct, 26). *Crea tu primera aplicación Android desde Ubuntu (1st ed.)* [Online]. Available: <http://www.nosinmiubuntu.com/2011/10/crea-tu-primer-a-aplicacion-android.html>
- [4ndroid, 2011] 4ndroid (2011, May, 5). *Android: paso a paso (II) (1st ed.)* [Online]. Available: <http://4ndroid.com/android-paso-a-paso-ii/>
- [Apple, 2010] Apple Inc. (2010). *iOS*. [Online]. Available: <https://www.apple.com/ios/>
- [Code Factory, 2013] Code Factory, S.L. (2013). *Enlarge your horizons with Mobile Magnifier (1st ed.)* [Online]. Available: <http://www.codefactory.es/en/^+products.asp?id=312>
- [Cruz., 2008] R. A. Cruz. (2008, May, 3). *Android (1st ed.)* [Online]. Available: <http://www.utm.mx/~caff/poo/AndroidIntro.pdf>
- [Google Play, 2011] Google Play (2011). *Magnify*. [Online]. Available: https://play.google.com/store/apps/details?id=com.appdlab.magnify&hl=en_GB
- [Google Play, 2012] Google Play (2012, Dec, 27). *Cozy Magnifier & Microscope + (2nd ed.)* [Online]. Available: https://play.google.com/store/apps/details?id=com.hantor.CozyMag&feature=search_result#?t=W251bGwsMSwxLDEsImNvbS5oYW50b3luQ296eU1hZyJd
- [Google, 2008] Google (2008). *Android*. [Online]. Available: <http://www.android.com/>
- [insensiv, 2009] insensiv GmbH. (2009, Nov, 25). *iLoupe*. [Online]. Available: <https://itunes.apple.com/au/app/iloupe/id327533744?mt=8&ign-mpt=uo%3D4>
- [Issue, 2007] Issue (2007, Jul). *Expanding the View: A Review of Mobile Magnifier and ZOOMS Screen Magnifiers for Cell Phones (Vol 8.Nº4 ed.)* [Online]. Available: <http://www.afb.org/afbpress/pub.asp?docid=aw080404>
- [Itseez., 2012] OpenCV (2012). *Android (1st ed.)* [Online]. Available: <http://opencv.org/android>
- [Leibs., 2010] A. Leibs (2010). *Zoom: Apple's Built-in Screen Magnifier (1st ed.)* [Online]. Available: <http://assistivetechology.about.com/od/ATCAT10/p/Zoom-Apples-Built-In-Screen-Magnifier.htm>
- [Microsoft, 2010] Microsoft (2010). *Windows Phone*. [Online]. Available: <http://www.windowsphone.com/en-us>

[Unknown, 2012a] OpenCV dev. team (2012, Dec, 26). *Introduction (1st ed.)* [Online]. Available: <http://docs.opencv.org/modules/core/doc/intro.html>

[Vico, 2012] A. J. Vico (2011, Feb, 17). *Arquitectura de Android (1st ed.)* [Online]. Available: <http://columna80.wordpress.com/2011/02/17/arquitectura-de-android/>

[Wikipedia, 2012a] Wikipedia (2012, Mar, 7). *Mobile Speak (last ed.)* [Online]. Available: http://en.wikipedia.org/wiki/Mobile_Speak

[Wikipedia, 2012b] Wikipedia (2012, Oct, 7). *OpenCV (last ed.)* [Online]. Available: <http://es.wikipedia.org/wiki/OpenCV>

[Wikipedia, 2013] Wikipedia (2013, Jan, 3). *Android (last ed.)* [Online]. Available: <http://es.wikipedia.org/wiki/Android>

8. Anexo: Manual de usuario de la librería

Este apartado va dirigido especialmente para la persona que implementará la interfaz en la aplicación, en el cual se va a describir cómo se ha preparado el entorno de trabajo, algunas recomendaciones y consejos. Para descripción de cada función se ha usado el Javadoc preparado para tal fin (que no se incluye aquí).

8.1. Preparación del entorno de trabajo

La librería está desarrollada en Java usando Eclipse trabajando en Ubuntu 10.0. Los pasos que deberían seguirse para preparar el entorno están en:

<http://developer.android.com/sdk/installing/installing-adt.html>

Una vez instalado el plugin de Eclipse y el SDK de Android. Hay que incorporar el paquete OpenCV. El tutorial de instalación y el propio paquete se pueden obtener en:

http://docs.opencv.org/doc/tutorials/introduction/android_binary_package/dev_with_OpenCV_on_Android.html#dev-with-ocv-on-android

Cuando ya se tenga el OpenCV en el Eclipse. Se recomienda probar los ejemplos y tutoriales de programas que usan OpenCV para ver sus capacidades y cómo funciona en general.

Hay un pequeño problema con la versión de Android 4.0 y Ubuntu. Al conectar un móvil al PC y ejecutar un ejemplo, Eclipse no va a reconocer directamente al dispositivo. La solución a este problema se puede encontrar en:

<http://developer.android.com/tools/device.html#setting-up>

8.2. Cómo trabajar

Para poder probar cualquier aplicación que utilice OpenCV, se recomienda usar la opción de Importar (o crear el workspace donde se encuentran los proyectos que se quiere incorporar en Eclipse), seleccionar Android y “Existing Android Code into Workspace”, seleccionar el proyecto e importarlo al Eclipse. Hay que añadir que haga uso de la librería OpenCV para evitar los errores (clic con el botón derecho sobre el proyecto, seleccionar “Properties”, en la ventana que aparezca, “Android”, y abajo en “Library” añadir el OpenCV, luego seleccionar “Java Build Path”, la pestaña de “Projects” y “Add” para seleccionar el proyecto con la librería OpenCV).

Cuando los errores desaparezcan, se puede ejecutar la aplicación (click en “Run”) sobre un emulador de Android o un dispositivo móvil. Al trabajar con la cámara hay que usar un dispositivo físico.

Una vez teniendo el proyecto Magnificador en Eclipse, es necesario borrar el contenido del fichero MagnificadorActivity.java y dejar solamente estas líneas:

```
package es.upm.fi.magnificador;

import org.opencv.android.BaseLoaderCallback;
import org.opencv.android.LoaderCallbackInterface;
import org.opencv.android.OpenCVLoader;

import android.app.Activity;
import android.app.AlertDialog;
import android.content.DialogInterface;
import android.os.Bundle;
import android.util.Log;
import android.view.Menu;
import android.view.MenuItem;
import android.view.Window;

public class MagnificadorActivity extends Activity {
    private static final String TAG = "MagnificadorActivity";

    private MagnificadorProcess mView;

    private BaseLoaderCallback mOpenCVCallBack = new BaseLoaderCallback(this) {
        @Override
        public void onManagerConnected(int status) {
            switch (status) {
                case LoaderCallbackInterface.SUCCESS:
                {
                    Log.i(TAG, "OpenCV loaded successfully");
                    // Create and set View
                    mView = new MagnificadorProcess(mAppContext);
                    setContentView(mView);
                    // Check native OpenCV camera
                    if( !mView.openCamera() ) {
                        AlertDialog ad = new AlertDialog.Builder(mAppContext).create();
                        ad.setCancelable(false); // This blocks the 'BACK' button
                        ad.setMessage("Fatal error: can't open camera!");
                        ad.setButton("OK", new DialogInterface.OnClickListener() {
                            public void onClick(DialogInterface dialog, int which) {
                                dialog.dismiss();
                                finish();
                            }
                        });
                        ad.show();
                    }
                }
                break;
            default:
            {
                super.onManagerConnected(status);
            } break;
            }
        }
    };
};
```

```

    public Sample2NativeCamera() {
        Log.i(TAG, "Instantiated new " + this.getClass());
    }

    @Override
    protected void onPause() {
        Log.i(TAG, "onPause");
        super.onPause();
        if (null != mView)
            mView.releaseCamera();
    }

    @Override
    protected void onResume() {
        Log.i(TAG, "onResume");
        super.onResume();
        if((null != mView) && !mView.openCamera() ) {
            AlertDialog ad = new AlertDialog.Builder(this).create();
            ad.setCancelable(false); // This blocks the 'BACK' button
            ad.setMessage("Fatal error: can't open camera!");
            ad.setButton("OK", new DialogInterface.OnClickListener() {
                public void onClick(DialogInterface dialog, int which) {
                    dialog.dismiss();
                    finish();
                }
            });
            ad.show();
        }
    }

    /** Called when the activity is first created. */
    @Override
    public void onCreate(Bundle savedInstanceState) {
        Log.i(TAG, "onCreate");
        super.onCreate(savedInstanceState);
        requestWindowFeature(Window.FEATURE_NO_TITLE);
        Log.i(TAG, "Trying to load OpenCV library");
        if (!OpenCVLoader.initAsync(OpenCVLoader.OPENCV_VERSION_2_4_2, this,
                                   mOpenCVCallback))
        {
            Log.e(TAG, "Cannot connect to OpenCV Manager");
        }
    }

```

Al sustituir el contenido de este fichero, se podrá a implementar una nueva interfaz en este mismo fichero o creando una clase nueva. Estas líneas de código son imprescindibles para cargar la librería OpenCV e inicializar el funcionamiento de la cámara.

El acceso a los métodos del magnificador se hace mediante el objeto mView. Para conocer las funciones de cada una de ellas consulte el Javadoc.

Este documento esta firmado por



Firmante	CN=tfgm.fi.upm.es, OU=CCFI, O=Facultad de Informatica - UPM, C=ES
Fecha/Hora	Tue Feb 11 23:12:32 CET 2014
Emisor del Certificado	EMAILADDRESS=camanager@fi.upm.es, CN=CA Facultad de Informatica, O=Facultad de Informatica - UPM, C=ES
Numero de Serie	630
Metodo	urn:adobe.com:Adobe.PPKLite:adbe.pkcs7.sha1 (Adobe Signature)